

BỘ GIÁO DỤC VÀ ĐÀO TẠO



GIÁO TRÌNH KỸ THUẬT SỐ

DÙNG CHO CÁC TRƯỜNG ĐÀO TẠO HỆ TRUNG CẤP CHUYÊN NGHIỆP



NHÀ XUẤT BẢN GIÁO DỤC

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TS. NGUYỄN VIỆT NGUYÊN

GIÁO TRÌNH

KỸ THUẬT SỐ

Dùng cho các trường đào tạo hệ Trung cấp chuyên nghiệp

(Tái bản lần thứ nhất)

NHÀ XUẤT BẢN GIÁO DỤC

Bản quyền thuộc HEVOBCO – Nhà xuất bản Giáo dục

04 – 2008/CXB/349 – 1999/GD

Mã số : 7K559y8 – DAI

Lời giới thiệu

Từ nhiều năm, giáo trình đào tạo nhân lực trình độ Trung cấp chuyên nghiệp chưa đáp ứng được yêu cầu chất lượng và chưa phù hợp với nhịp độ phát triển của đất nước.

Mặc dù luật Giáo dục đã quy định, Hiệu trưởng các trường quyết định giáo trình dạy của trường mình. Tuy nhiên, do kinh phí có hạn, trình độ đội ngũ cán bộ giảng viên không đồng đều, vì vậy cùng một môn học nhưng nội dung và dung lượng kiến thức giảng dạy ở mỗi trường một khác.

Để giúp các trường từng bước có giáo trình phục vụ việc giảng dạy và học tập tốt hơn đồng thời giúp học sinh sau khi tốt nghiệp dù được đào tạo ở đâu nhưng cũng có kiến thức chung như nhau, Bộ Giáo dục và Đào tạo đã tổ chức biên soạn các giáo trình:

1. Giáo trình Kỹ thuật số
2. Giáo trình Kỹ thuật điện
3. Giáo trình Cơ kỹ thuật
4. Giáo trình Công nghệ hàn
5. Giáo trình Kỹ thuật nguội

Tác giả biên soạn những giáo trình này là các nhà giáo có trình độ chuyên môn tốt và giàu kinh nghiệm giảng dạy.

Để nâng cao chất lượng và tính sư phạm của giáo trình, Bộ Giáo dục và Đào tạo đã ra quyết định số 2444/QĐ- BGDDT ngày 18 tháng 5 năm 2006 về việc thành lập Hội đồng thẩm định cho các môn trên.

Thực hiện quyết định của Bộ trưởng Bộ Giáo dục và Đào tạo, các thành viên trong Hội đồng thẩm định đã làm việc nghiêm túc và cùng với tác giả chỉnh sửa để nâng cao chất lượng, phù hợp với trình độ của cấp đào tạo.

Những nội dung kiến thức cơ bản trong giáo trình cần được dạy và học thống nhất trên toàn quốc khi trường có chuyên ngành đào tạo giảng dạy môn học này. Vì vậy các trường cần cung ứng đầy đủ giáo trình này cho giáo viên và học sinh.

Tuỳ theo nhu cầu cụ thể của từng trường, các trường có thể sử dụng 70% dung lượng của giáo trình và tự soạn thêm 30% dung lượng của môn học cho phù hợp với yêu cầu đào tạo nguồn nhân lực của địa phương.

Trong quá trình dạy và học, các trường phát hiện thấy sai sót hoặc có những nội dung cần điều chỉnh xin góp ý để các cuốn giáo trình hoàn thiện hơn. Mọi ý kiến góp ý xin gửi về Vụ Giáo dục chuyên nghiệp - Bộ Giáo dục và Đào tạo - 49 Đại Cồ Việt, Hà Nội hoặc Công ty Cổ phần sách Đại học - Dạy nghề - 25 Hàn Thuyên, Hà Nội.

VỤ GDCN

Mở đầu

Giáo trình *Kỹ thuật số* được biên soạn theo đề cương do vụ Giáo dục chuyên nghiệp, Bộ Giáo dục & Đào tạo xây dựng và thông qua. Nội dung được biên soạn theo tinh thần ngắn gọn, dễ hiểu. Các kiến thức trong toàn bộ giáo trình có mối liên hệ logic, chặt chẽ. Tuy vậy, giáo trình cũng chỉ là một phần trong nội dung của chuyên ngành đào tạo cho nên người dạy, người học cần tham khảo thêm các giáo trình có liên quan đối với ngành học để việc sử dụng giáo trình có hiệu quả hơn.

Khi biên soạn giáo trình, chúng tôi đã cố gắng cập nhật những kiến thức mới có liên quan đến môn học và phù hợp với đối tượng sử dụng cũng như cố gắng gắn những nội dung lý thuyết với những vấn đề thực tế thường gặp trong sản xuất, đời sống để giáo trình có tính thực tiễn cao.

Nội dung của giáo trình được biên soạn gồm 5 chương :

Chương 1. Cơ sở kỹ thuật số ; Chương 2. Các cổng logic cơ bản và mạch điện cổng ; Chương 3. Các mạch logic tổ hợp ; Chương 4. Triggers và các phần tử logic dãy ; Chương 5. Các bộ nhớ bán dẫn;

Trong quá trình sử dụng, tùy theo yêu cầu cụ thể có thể điều chỉnh số tiết trong mỗi chương. Trong giáo trình, chúng tôi không đề ra nội dung thực tập của từng chương, vì trong thiết bị phục vụ cho thực tập của các trường không đồng nhất. Vì vậy, căn cứ vào trang thiết bị đã có của từng trường và khả năng tổ chức cho học sinh thực tập ở các xí nghiệp bên ngoài mà trường xây dựng thời lượng và nội dung thực tập cụ thể. Thời lượng thực tập tối thiểu nói chung cũng không ít hơn thời lượng học lý thuyết của môn học.

Giáo trình được biên soạn cho đối tượng là học sinh Trung cấp chuyên nghiệp, Công nhân lành nghề bậc 3/7 đồng thời cũng là tài liệu tham khảo bổ ích cho sinh viên Cao đẳng kỹ thuật cũng như Kỹ thuật viên đang làm việc ở các cơ sở kinh tế trên nhiều lĩnh vực khác nhau.

Mặc dù đã cố gắng nhưng chắc chắn không tránh khỏi khiếm khuyết. Rất mong nhận được ý kiến đóng góp của người sử dụng để lần tái bản sau được hoàn chỉnh hơn.

TÁC GIẢ

Chương 1

CƠ SỞ KỸ THUẬT SỐ

1.1. CÁC HỆ THỐNG SỐ ĐẾM

1.1.1. Các đại lượng vật lý được theo dõi, đo lường, ghi lại, tính toán... cần được biểu diễn bằng giá trị thực của chúng một cách chính xác để thuận lợi trong việc xử lý có hiệu quả. Có hai cách biểu diễn các đại lượng này :

1. Biểu diễn ở dạng tương tự : khi hàm biểu diễn là đại lượng biến thiên liên tục theo thời gian với cùng một cách thức ta có tín hiệu tương tự hay tín hiệu analog mô tả biểu diễn đại lượng cần xử lý, ví dụ như hiệu điện thế ở đầu ra của một micro có thể biến thiên liên tục trong khoảng giá trị từ 0 tới khoảng 100mV, biểu diễn tiếng nói của người đang sử dụng micro, hoặc kim đồng hồ đo tốc độ sẽ biến thiên liên tục khi một chiếc ô tô đang chạy để biểu diễn tốc độ của ô tô trong khoảng từ 0 đến 50m/s...

2. Biểu diễn đại lượng ở dạng số : khi đó hàm biểu diễn sẽ biến thiên không liên tục theo thời gian và người ta dùng các ký tự bằng số để mô tả biểu diễn nó, ta nhận được tín hiệu số hay tín hiệu digital với đặc trưng là sự biến thiên từng bước rời rạc.

Tương ứng với điều trên, một mạch điện tử, một thiết bị hay hệ thống điện tử làm nhiệm vụ xử lý các tín hiệu thuộc loại nào sẽ mang tên của loại đó (ví dụ hệ thống tương tự và hệ thống số). Nhìn chung thế giới hiện thực xung quanh ta là thế giới tương tự, tức là các đại lượng xung quanh ta có bản chất là tương tự tác động tới đầu vào và yêu cầu xuất hiện ở đầu ra một hệ thống gia công xử lý tín tức. Kỹ thuật xử lý số tín hiệu dùng các hệ thống số như vậy có vai trò trung gian trong ba bước :

* Biến đổi đại lượng đầu vào tự nhiên dạng tương tự thành tín hiệu số.

* Xử lý thông tin tín hiệu số vừa nhận được.

* Biến đổi ở cổng ra từ tín hiệu dạng số về lại dạng tương tự.

Nguyên nhân của việc làm qua bước trung gian xử lý tín hiệu số xuất phát từ :

- Thói quen từ bản chất của con người "số hoá" các đại lượng cần quan tâm xử lý, ví dụ như khi ta nói nhiệt độ phòng là 25°C thực ra chỉ là con số gần đúng đã được làm tròn của giá trị thực đang có.

- Kỹ thuật xử lý số thể hiện nhiều ưu điểm vượt trội so với các phương pháp xử lý truyền thống trước đây : dễ dàng hơn trong thiết kế, thuận lợi trong lưu giữ thông tin theo thời gian, tính chính xác và độ tin cậy đạt được cao, có thể lập trình để xử lý tự động, ít chịu ảnh hưởng của tác động lạ (nhiều)...

3. Quá trình biến đổi một tín hiệu dạng tương tự sang dạng tín hiệu số cần 3 bước cơ bản sau đây :

* Thực hiện việc rời rạc hoá tín hiệu tương tự bằng cách lấy mẫu các giá trị của nó ở những thời điểm xác định. Bước này cần chú ý làm giảm tới mức ít nhất việc mất mát thông tin, muốn vậy chu kỳ (nhịp) lấy mẫu phải mau hơn khoảng hơn hai lần chu kỳ mau nhất của tín hiệu ($f_{\text{mẫu}} \geq 2f_{\text{max}}$).

* Thực hiện việc làm tròn (lượng tử hoá) các giá trị mẫu đã lấy. Muốn vậy cần chọn ra một đơn vị rời rạc nhỏ nhất về độ lớn gọi là 1 bước (1 giá trị) lượng tử cùng đơn vị đo lường với giá trị đã rời rạc ở trên và đánh giá chúng bằng bao nhiêu lần phần nguyên giá trị lượng tử.

* Thực hiện việc biểu diễn các giá trị vừa làm tròn thành các ký số trong một hệ thống số đếm được lựa chọn, ví dụ trong hệ thập phân (hệ đếm mười) hay trong hệ đếm nhị phân (hệ đếm hai), công việc này gọi là mã hoá các giá trị làm tròn đã chọn.

1.1.2. Để thực hiện việc mã hoá, phải sử dụng một hệ thống số đếm nào đó. Tính chất quan trọng nhất của một hệ thống số đếm là sử dụng một dãy các ký tự để thể hiện một con số trong hệ. Giá trị con số thể hiện qua giá trị và vị trí của mỗi ký số, vị trí này có trọng lượng (trọng số) tăng dần khi dịch vị trí từ phải qua trái. Trong kỹ thuật số có 4 hệ thống số đếm quan trọng được sử dụng :

1. Hệ đếm thập phân (hệ mười) gồm 10 ký tự các số tự nhiên từ 0 đến 9 ; vị trí của chúng thể hiện hàng đơn vị, hàng chục, hàng trăm... tính từ tận cùng bên phải (LSD) sang tận cùng bên trái (MSD) với phần nguyên và ngược lại từ trái qua phải là phần chục, phần trăm... với phần lẻ sau dấu phẩy như ta thường gặp hàng ngày.

Ví dụ: 181,25 là một số thập phân gồm 2 phần : phần nguyên 181 và phần lẻ 0,25 được biểu diễn ở dạng:

$$181,25 = 1.10^2 + 8.10^1 + 1.10^0 + 2.10^{-1} + 5.10^{-2}$$

nghĩa là giá trị của một số luôn bằng tổng các tích giữa giá trị của mỗi ký số (chữ số) với giá trị vị trí (trọng số thập phân) của nó. Điều này tương tự với mọi hệ thống số đếm khác.

2. Hệ đếm nhị phân (hệ hai) gồm 2 ký số (2 chữ số) tự nhiên là 0 và 1 và cũng tuân theo luật vị trí biểu diễn trọng số tương ứng (hay cấp nhị phân mà nó thể hiện).

Một số nhị phân n cấp (gọi là n bit nhị phân) ở hệ mười có dạng:

$$A_{(10)} = a_{n-1}.2^{n-1} + a_{n-2}.2^{n-2} + \dots + a_1.2^1 + a_0.2^0 \quad (1.1)$$

trong đó : a_{n-1} gọi là bit có nghĩa lớn nhất (MSB hay bit già nhất)

a_0 là bit có nghĩa nhỏ nhất (LSB hay bit trẻ nhất)

Các ký tự a_k chỉ nhận một trong hai giá trị 0 hoặc 1.

Ví dụ : Với số nhị phân $10110101_{(2)}$ có $n = 8$ (8 bit hay thường gọi là 1byte) ở hệ đếm mười nó biểu diễn số:

$$\begin{aligned} & 1.2^7 + 0.2^6 + 1.2^5 + 1.2^4 + 0.2^3 + 1.2^2 + 0.2^1 + 1.2^0 \\ & = 128 + 0 + 32 + 16 + 0 + 4 + 0 + 1 = 181_{(10)} \end{aligned}$$

Một số nhị phân có thể gồm 2 phần : bên trái dấu phẩy là phần nguyên sử dụng hệ thức (1.1) để xác định biểu diễn trong hệ mười. Nếu các ký số 0, 1 nằm bên phải, sau dấu phẩy chúng sẽ biểu diễn phần lẻ, được biểu diễn trong hệ 10 tương đương như sau:

$$\begin{aligned} 0,1010_{(2)} &= 1.2^{-1} + 0.2^{-2} + 1.2^{-3} + 0.2^{-4} \\ &= \frac{1}{2} + 0 + \frac{1}{8} + 0 \\ &= 0,625_{(10)} \end{aligned} \quad (1.1)'$$

Như vậy số nhị phân $10110101,1010_{(2)} = 181,625_{(10)}$

3. Hệ đếm mười sáu (hệ HEXA) sử dụng 16 ký tự là 10 chữ số tự nhiên và 6 chữ cái viết hoa đầu tiên:

0 1 2 3 4 5 6 7 8 9 A B C D E F

vị trí các ký tự thể hiện trọng số $16^i_{(10)}$ ($i = 0, 1, 2, \dots, 9$) của nó.

Ví dụ : Các số $356_{(16)}$, $2AF_{(16)}$ hay $1BC2_{(16)}$ (biểu diễn trong hệ mười sáu) có giá trị tương đương trong hệ mười là :

$$\begin{aligned} 356_{(16)} &= 3.16^2 + 5.16^1 + 6.16^0 \\ &= 768 + 80 + 6 = 854_{(10)} \end{aligned}$$

$$\begin{aligned} 2AF_{(16)} &= 2.16^2 + 10.16^1 + 15.16^0 \\ &= 512 + 160 + 15 = 687_{(10)} \end{aligned}$$

$$1BC2_{(16)} = 7106_{(10)}$$

Một điều cần chú ý là mỗi ký tự chữ cái A, B, C, D, E hay F trong hệ đếm mười sáu đại diện cho một nhóm 4 ký số tương đương trong hệ đếm hai, nhờ đó có thể thu gọn cách viết hệ hai :

$$A_{(16)} = 1010_{(2)} ; \quad D_{(16)} = 1101_{(2)}$$

$$B_{(16)} = 1011_{(2)} ; \quad E_{(16)} = 1110_{(2)}$$

$$C_{(16)} = 1100_{(2)} ; \quad F_{(16)} = 1111_{(2)}$$

4. Hệ đếm tám (OCTA) sử dụng 8 số tự nhiên đầu tiên 0, 1, 2, 3, 4, 5, 6, 7 và cũng theo luật vị trí để xác định trọng số thập phân 8^k ($k = 0, 1, \dots, 9$) của mỗi ký số :

----	8^4	8^3	8^2	8^1	8^0	8^{-1}	8^{-2}	8^{-3}	8^{-4}	----
------	-------	-------	-------	-------	-------	----------	----------	----------	----------	------

dấu chấm hay dấu phẩy

$$\begin{aligned} \text{Ví dụ : } 375_{(8)} &= 3.8^2 + 7.8^1 + 5.8^0 \\ &= 3.64 + 7.8 + 5.1 = 253_{(10)} \end{aligned}$$

$$\text{hay } 25.4_{(8)} = 2.8^1 + 5.8^0 + 4.8^{-1} = 21.5_{(10)}$$

Bảng 1.1 biểu diễn 16 số tự nhiên đầu tiên trong các hệ đếm vừa trình bày ở trên, để tránh nhầm lẫn người ta thường ký hiệu hệ đếm ở phía dưới tận cùng bên phải của con số đã cho như đã làm trong các ví dụ trên.

Bảng 1.1. ĐỐI CHIẾU 16 SỐ TỰ NHIÊN ĐẦU TIÊN TRONG 4 HỆ ĐẾM THƯỜNG DÙNG

Hệ mười	Hệ 2	Hệ 8	Hệ 16
0	0 0 0 0	0	0
1	0 0 0 1	1	1
2	0 0 1 0	2	2
3	0 0 1 1	3	3
4	0 1 0 0	4	4
5	0 1 0 1	5	5
6	0 1 1 0	6	6
7	0 1 1 1	7	7
8	1 0 0 0	10	8
9	1 0 0 1	11	9
10	1 0 1 0	12	A
11	1 0 1 1	13	B
12	1 1 0 0	14	C
13	1 1 0 1	15	D
14	1 1 1 0	16	E
15	1 1 1 1	17	F

1.1.3. Để biểu diễn 1 con số trong một hệ đếm, ví dụ trong hệ thập phân, ta sử dụng 4 phần:

- * Phần dấu + hay -
- * Phần nguyên đứng trước dấu phẩy (.)
- * Dấu phẩy phân cách phần nguyên với phần lẻ
- * Phần lẻ nằm sau dấu phẩy

Ví dụ : - 125,258

Để thuận tiện cho việc thao tác tính toán người ta thường sử dụng phương pháp dấu phẩy tính, nội dung là đặt dấu phẩy sau chữ số tận cùng bên phải gọi là biểu diễn ở dạng chỉ có phần nguyên:

$$125,258 = 125\ 258, (x\ 10^{-3})$$

hoặc đặt dấu phẩy trước chữ số tận cùng bên trái gọi là cách biểu diễn ở dạng chỉ có phần lẻ

$$\text{Ví dụ : } 18,36_{10} = ,1836. 10^2$$

Phương pháp dịch dấu phẩy giúp việc thiết kế mạch thực hiện đơn giản hơn.

* Để biểu diễn một số có dấu (dương hay âm) trong hệ nhị phân có thể sử dụng cách bổ sung vào số đã cho một ký số (được gọi là bit thể hiện dấu) ở đầu phía trước của số đã cho theo quy định:

Ký số 1 biểu diễn số nhị phân sau nó là số âm

Ký số 0 biểu diễn số nhị phân tiếp sau là số dương

ta gọi đây là cách biểu diễn dấu và trị số thật để phân biệt với cách biểu hiện dấu khác (như dùng số bù 2 sẽ xét tới ở sau).

$$\text{Ví dụ: } -242_{(10)} = 1\ 11110010_{(2)} \quad ; \quad +150_{(10)} = 0\ 10010110_{(2)}$$

* Nếu dùng một số bù 2 để biểu diễn một số nhị phân dấu âm (việc này sẽ thuận lợi hơn trong các thiết bị tính toán so với cách dùng trị số thật như trình bày ở trên) ta sẽ bắt đầu bằng thiết lập số bù 1 (đảo) của số đã cho, sau đó cộng thêm 1 vào số bù 1 vừa tạo ra ta sẽ nhận được số nhị phân bù 2 của số nhị phân ban đầu.

Ví dụ:

$$\begin{array}{r} 45_{10} = 1\ 0\ 1\ 1\ 0\ 1_2 \\ \downarrow\downarrow\downarrow\downarrow\downarrow \\ 0\ 1\ 0\ 0\ 1\ 0 \text{ bù 1} \end{array}$$

có bù 1 bằng cách đảo toàn bộ các bit của số nhị phân ban đầu, sau đó cộng thêm 1 vào kết quả ta có: $010010 + 1 = 010011$ chính là dạng biểu diễn số bù 2 của số nhị phân đã cho ban đầu 101101 . Khi đó số có dấu được thể hiện theo quy định là:

$$\begin{array}{l} \text{trị số thật} \\ \left\{ \begin{array}{l} 0\ \overline{101101}_2 = +45_{10} \\ 1\ \underline{010011}_2 = -45_{10} \end{array} \right. \\ \text{bit dấu} \end{array}$$

Dạng bù 2 của trị số thật 101101 .

1.2. ĐẠI CƯƠNG VỀ CÁC PHÉP TÍNH SỐ HỌC TRONG HỆ NHỊ PHÂN

Các phép tính số học trong hệ nhị phân sẽ được đề cập chi tiết hơn ở các giáo trình liên quan khi xét tới các linh kiện số thực hiện các phép tính này.

Tuy nhiên, để có những khái niệm ban đầu thuần túy là các cơ sở toán học nhị phân, trong tiết này chúng ta sẽ tóm tắt một số quy tắc chính thực hiện các phép cộng, trừ, nhân và chia các số trong hệ hai, khía cạnh kỹ thuật mạch điện tử số thực hiện chúng sẽ chưa được đề cập tới vì kỹ thuật thực hiện có những điểm khác hơn.

1.2.1. Phép cộng nhị phân được tiến hành giống như phép cộng trong hệ thập phân tức là phải cộng các ký số cùng cột (cùng trọng số nhị phân) theo quy tắc (1.2) :

$$\left. \begin{array}{l} 0 + 0 = 0 \\ 0 + 1 = 1 + 0 = 1 \\ 1 + 1 = 1 \ 0 = 0 + \text{nghé 1 cộng vào cột tiếp bên trái} \\ 1 + 1 + 1 = 1 \ 1 = 1 + \text{nghé 1 cộng vào cột tiếp bên trái} \end{array} \right\} (1.2)$$

Khi phép cộng không có nghé, sử dụng hai quy tắc đầu trong (1.2)

Khi phép cộng có nghé sử dụng hai quy tắc sau trong hệ thức (1.2)

Khi cộng nhiều số nhị phân ta thực hiện cộng dần (từng hai số một).

Ví dụ : các phép cộng nhị phân (ta có ghi chú số thập phân tương đương bên cạnh để dễ so sánh).

$$\begin{array}{r} 0 \ 1 \ 1 \ (3) \\ + \ 1 \ 1 \ 0 \ (6) \\ \hline 1 \ 0 \ 0 \ 1 \ (9) \end{array} \quad \begin{array}{r} 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ (45) \\ + \ 1 \ 1 \ 1 \ 1 \ (15) \\ \hline 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ (60) \end{array} \quad \begin{array}{r} 1 \ 1, \ 0 \ 1 \ 1 \ (3,375) \\ + \ 1 \ 0, \ 1 \ 1 \ 0 \ (2,750) \\ \hline 1 \ 1 \ 0, \ 0 \ 0 \ 1 \ (6,125) \end{array}$$

1.2.2. Phép trừ nhị phân thực hiện theo quy tắc (1.3) với hai ký số (bit) nhị phân cùng trọng số.

$$\left. \begin{array}{l} 0 - 0 = 0 \\ 1 - 0 = 1 \\ 1 - 1 = 0 \\ 10 - 1 = 1 \text{ (vay 1 ở cấp cao hơn)} \end{array} \right\} (1.3)$$

Chú ý : khi ở cột thứ k xảy ra việc $(0 - 1)$ ta áp dụng quy tắc vay 1 ở cấp $(k+1)$ thực hiện theo quy tắc hàng cuối của (1.3) với số chỉ có duy nhất bit MSB bằng 1 $(100 \dots 0)$ sau khi trừ đi số nhị phân chỉ có duy nhất một bit LSB kết quả sẽ là số bù 1 của số bị trừ :

$$\underbrace{1 \ 0 \ 0 \ \dots \ 0 \ 0}_{n \text{ số } 0} - 1 = 0 \ \underbrace{1 \ 1 \ \dots \ 1 \ 1}_{n \text{ số } 1}$$

Ví dụ:

$$\begin{array}{r}
 \begin{array}{ccccccc}
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & (199) \\
 - & 1 & 1 & 0 & 1 & 1 & 0 & 1 & (109) \\
 \hline
 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & (90)
 \end{array}
 \end{array}$$

Cần chú ý rằng có thể thực hiện phép trừ nhờ đổi dấu số trừ sau đó dùng các quy tắc của phép cộng để thực hiện giữa số bị trừ (số hạng đầu) và số trừ sau khi đã đổi dấu (số hạng sau).

1.2.3. Phép nhân hai số nhị phân thực hiện giống phép nhân trong hệ mười theo quy tắc (1.4) (phép nhân không dấu).

$$\left. \begin{array}{l}
 0 \cdot 0 = 0 \\
 1 \cdot 0 = 0 \cdot 1 = 0 \\
 1 \cdot 1 = 1
 \end{array} \right\} \quad (1.4)$$

Thực hiện nhân liên tiếp từng cột của một thừa số với tất cả các cột của thừa số kia, hai hàng kết quả của 2 cột liên tiếp nhau phải đặt dịch trái 1 nhị, sau cùng cộng các tích (các hàng tích) lại sẽ có tích toàn phần.

$$\begin{array}{r}
 1010 \leftarrow \text{Số bị nhân (10)} \\
 101 \leftarrow \text{Số nhân (5)} \\
 \hline
 \begin{array}{r}
 1010 \\
 + 0000 \\
 1010 \\
 \hline
 110010 (50)
 \end{array}
 \end{array}$$

các hàng tích số bán phần

tích số toàn phần

Khi số bị nhân có m chữ số lẻ (sau dấu phẩy) và số nhân có n chữ số lẻ, ta bỏ dấu phẩy của cả hai thừa số và thực hiện quy tắc nhân như với hai số nhị phân chỉ có phần nguyên. Kết quả ở tích số toàn phần, dấu phẩy được đặt ở vị trí trước cột thứ (m+n) tính từ phải qua trái như vẫn làm với hệ mười.

1.2.4. Phép chia nhị phân khó thực hiện hơn. Có thể tách thành hai trường hợp đặc trưng:

1. Khi số bị chia lớn hơn số chia

* **Bước 1:** Bắt đầu thực hiện từ 1 nhóm các cột có trọng số cao nhất (phần trái) của số bị chia có số ký số trong nhóm bằng hoặc lớn hơn một số lượng ký số của số chia. Thương số sẽ bằng 1 khi chia được và sẽ bằng 0 khi không chia được.

* **Bước 2:** Khi phép chia với nhóm trên thực hiện được, cần thực hiện phép trừ để lấy phần dư (giữa nhóm bị chia và số chia). Sau đó lần lượt hạ các cột tiếp sau chưa sử dụng tới của số bị chia. Mỗi lần hạ một cột, thương số được điền thêm ký số 1 nếu nhóm bị chia mới lớn hơn số bị chia hoặc điền thêm ký số 0 (phía phải của thương) nếu sau mỗi lần hạ, nhóm bị chia mới vẫn còn nhỏ hơn số bị chia... cứ làm tiếp tục cho tới khi hết.

* **Bước 3:** Nếu phép chia không hết (còn phần dư) sau khi đã hạ đến bit LSB, sẽ thực hiện lấy thêm phần lẻ của thương nếu có yêu cầu sau khi đã đặt dấu phẩy khi kết thúc phần nguyên. Muốn vậy thêm các số 0 ở sau phần số dư cuối cùng và tiến hành tiếp tục phép chia theo bước 1 và bước 2 cho tới lúc chia hết (phần dư bằng 0) hay đến số chữ số lẻ đủ theo yêu cầu.

Ví dụ 1: Thực hiện phép chia $101\ 101 : 101$

$$\begin{array}{r}
 101'101\ (45) \\
 - 101 \\
 \hline
 0001'01 \\
 - 000 \\
 \hline
 0010'1 \\
 - 000 \\
 \hline
 0101' \\
 - 101 \\
 \hline
 000 \quad \leftarrow \text{Phần dư}
 \end{array}
 \qquad
 \begin{array}{r}
 \overline{) 101\ (5)} \\
 \underline{1001} \quad \leftarrow \text{Thương số (9)}
 \end{array}$$

Ví dụ 2 : Thực hiện phép chia $110000 : 101\ (48 : 5)$

$$\begin{array}{r}
 110'000\ (48) \\
 \underline{101} \\
 0010'0'0' \\
 \underline{1\ 0\ 1} \\
 0\ 1\ 1\ 0 \\
 \underline{1\ 0\ 1} \\
 001 \quad \leftarrow \text{Phần dư}
 \end{array}
 \qquad
 \begin{array}{r}
 \overline{) 101\ (5)} \\
 \underline{1001,10\ (\approx 9,5)} \\
 \uparrow \\
 \text{Thương số lấy đến 1 số lẻ}
 \end{array}$$

2. Khi số bị chia nhỏ hơn số chia cách thực hiện giống như ví dụ trên, kết quả thương số chỉ có phần lẻ sau dấu phẩy, mỗi lần thêm một số 0 vào số bị chia cần ghi một ký số 0 vào thương số phía sau dấu phẩy cho tới khi số bị chia “lớn hơn” số chia, phép tính này tương tự như trong hệ mười.

1.3. CÁC PHÉP BIẾN ĐỔI SỐ BIỂU DIỄN TRONG CÁC HỆ THỐNG SỐ KHÁC NHAU

Hệ thập phân thường biểu diễn các đại lượng bên ngoài thiết bị số, do vậy để thực hiện việc giao tiếp với các thiết bị số, xuất hiện yêu cầu chuyển đổi một số từ hệ này sang hệ kia, đặc biệt với 4 hệ thống số thông dụng nhất : hệ mười, hệ hai, hệ tám và hệ mười sáu. Tiết này sẽ trình bày các quy tắc cơ bản để biến đổi một số đã cho đang ở hệ thống số này sang một hệ thống số mong muốn khác.

1.3.1. Chuyển đổi một số A từ biểu diễn nhị phân sang biểu diễn thập phân được thực hiện theo hệ thức đã biết:

$$A_{10} = a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \dots + a_1 2^1 + a_0 2^0 \quad (1.5)$$

Chú ý rằng vị trí của bit a_k có trọng số tương ứng 2^k .

Ví dụ : $A_2 = 101101$ khi chuyển sang A_{10} có biểu diễn tương đương theo (1.5) là:

$$\begin{aligned} A_{10} &= 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 32 + 0 + 8 + 4 + 0 + 1 = 45_{10} \end{aligned}$$

hoặc với 1 số nhiều bit hơn:

$$\begin{aligned} A_2 &= 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \\ A_{10} &= 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 231_{10} \end{aligned}$$

Nếu A_2 là một số nhị phân có phần nguyên và phần lẻ, phép chuyển đổi thực hiện theo hệ thức (1.1)' mở rộng.

$$\begin{aligned} \text{Ví dụ : } A_2 &= 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad , \quad 1 \quad 0 \quad \cdot \quad 1 \\ A_{10} &= 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ &= 45,625_{10} \end{aligned}$$

1.3.2. Chuyển đổi một số thập phân sang biểu diễn nhị phân được thực hiện riêng biệt cho phần nguyên và cho phần lẻ sau đó gộp hai kết quả lại.

1. Chuyển đổi phần nguyên có hai cách thực hiện

* Sử dụng biểu thức 1.1 ở dạng ngược với quá trình chuyển đổi hệ hai - mười : triển khai số thập phân (nguyên) thành tổng các lũy thừa của 2 sau đó xác định giá trị các ký tự (bit) a_k tương ứng.

Ví dụ minh họa các bước này:

Ví dụ 2: $A_{10} = 0,8325$ hãy tìm A_2 lấy tới 4 bit lẻ.

$A_{10} = 0,8325$	$2A_{10} = 1,665$	$2.0,665 = 1,33$	$0,33.2 = 0,66$	$0,66.2 = 1,32$
	$2A_{10}-1 = 0,665$	$1,33-1 = 0,33$	$0,66 < 1$	$1,32-1 = 0,32$
A_2	1	1	0	1

Vậy $A_2 = 0,1101$

Ví dụ 3: $A_{10} = 0,3125$ Tìm A_2 lấy tới 4 bit lẻ.

$A_{10} = 0,3125$	$2A_{10} = 0,625$	$0,625.2 = 1,25$	$0,25.2 = 0,5$	$0,5.2 = 1$
	$0,625 < 1$	$1,25 - 1 = 0,25$	$0,5 < 1$	$1 - 1 = 0$
A_2	0	1	0	1

Vậy $A_2 = 0,0101$

3. Nếu A_{10} có cả hai phần nguyên và phần lẻ: kết quả chung là sự kết hợp hai kết quả chuyển đổi riêng biệt như đã trình bày ở trên. Nếu sử dụng các ví dụ đã có với $A_{10} = 58,3125$ thì biểu diễn nhị phân của nó có dạng $A_2 = 111010,0101$.

1.3.3. Chuyển đổi số trong hệ 8 sang hệ khác (ví dụ từ A_8 sang A_{10}) cần sử dụng quy luật trọng số vị trí của các ký số đã cho để thực hiện.

$$\begin{aligned} \text{Ví dụ 1: } 375_8 &= 3 \cdot 8^2 + 7 \cdot 8^1 + 5 \cdot 8^0 \\ &= 3 \cdot 64 + 7 \cdot 8 + 5 \cdot 1 = 253_{10} \end{aligned}$$

$$\begin{aligned} \text{Ví dụ 2: } 74,6_8 &= 7 \cdot 8^1 + 4 \cdot 8^0 + 6 \cdot 8^{-1} \\ &= 60,75_{10} \end{aligned}$$

* Khi chuyển đổi một số A_{10} sang A_8 ta thực hiện chia liên tiếp cho 8 để lấy phần dư giống như quy luật đã làm với việc chuyển A_{10} sang A_2 .

Ví dụ 3: Cho $A_{10} = 268_{10}$ tìm A_8

$$\begin{array}{lcl} \frac{268}{8} = 33 + \text{div } 4 & \xrightarrow{\hspace{1cm}} & \\ \frac{33}{8} = 4 + \text{div } 1 & \xrightarrow{\hspace{1cm}} & \\ \frac{4}{8} = 0 + \text{div } 4 & \xrightarrow{\hspace{1cm}} & \end{array} \quad \begin{array}{l} \downarrow \\ \downarrow \\ \downarrow \end{array}$$

* Khi chuyển đổi một số A_8 sang A_2 thực hiện theo quy tắc đổi từng ký số của A_8 sang một nhóm gồm 3 ký số nhị phân tương đương theo bảng chuyển đổi dưới đây:

Ký số của A_8	0	1	2	3	4	5	6	7
nhóm nhị phân tương đương	000	001	010	011	100	101	110	111

Như vậy nếu A_8 gồm n ký số trong hệ tám (từ 0 tới 7) thì A_2 tương đương sẽ gồm 3. n bit biểu diễn.

Ví dụ với $A_8 = 472_8$; A_2 tương đương có dạng: $A_2 = 100\ 111\ 010_2$

hay với $A_8 = 1764_8$ thì A_2 tương đương có dạng $A_2 = 001\ 111\ 110\ 100_2$

* Khi cần đổi 1 số từ hệ hai A_2 sang số hệ tám A_8 ta thực hiện ngược lại bằng cách chia A_2 thành từng nhóm 3 bit và thay thế nhóm này bằng 1 ký số tương đương của hệ tám. Việc chia nhóm 3 bit bắt đầu tính từ bit trẻ nhất LSB. Nếu nhóm cuối cùng của A_2 chứa bit già nhất MSB không đủ 3 ký số thì thêm 0 vào trước bit MSB cho đủ.

Ví dụ : $A_7 = 101 \quad 110 \quad 010$

$$A_8 = \begin{pmatrix} 5 & 6 & 2 \end{pmatrix}$$

hay $A_2 = 001 \quad 011 \quad 110$

$$A_8 = \begin{pmatrix} 1 & & \\ & 3 & \\ & & 6 \end{pmatrix}$$

* Ta có nhận xét là việc chuyển đổi cách biểu diễn $A_2 \rightarrow A_8$ được thực hiện tương đối dễ dàng do đó nếu gặp một số A_2 khá lớn (ví dụ 24 bit) ta có thể viết tốc ký thu gọn A_2 bằng một số A_8 tương đương chỉ còn 8 ký số đặc biệt thuận lợi trong việc diễn đạt một số A_2 quá dài.

Ví dụ : Cho $A_{10} = 179$, hãy tìm biểu diễn tương đương của A_{10} qua A_8 và sau đó qua A_2

$$\begin{aligned}\frac{A_{10}}{8} &= \frac{179}{8} = 22 + \text{dư } 3 \\ \frac{22}{8} &= 2 + \text{dư } 6 \\ \frac{2}{8} &= 0 + \text{dư } 2\end{aligned}$$

Kết quả ta được A_8 tương đương $A_8 = 263_8$

Chuyển A_8 sang A_2 tương đương, ta nhận được kết quả $A_2 = 010110011_2$.

$$\begin{array}{rcccc} A_8 = & 2 & 6 & 3 \\ A_2 = & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{array}$$

Cách làm theo trình tự thao tác: $A_{10} \leftrightarrow A_8 \leftrightarrow A_2$ thường nhanh và thuận lợi trong kỹ thuật điện tử số hơn so với cách biến đổi trực tiếp $A_{10} \leftrightarrow A_2$ đã nêu trước.

1.3.4. Chuyển đổi từ một số hệ mười sáu sang hệ mười ($Z_{16} \rightarrow Z_{10}$)

Thực hiện theo quy tắc chung bằng ký tự sử dụng biểu diễn Z_{16} và quan tâm tới vị trí các ký tự của nó và trọng số tương ứng của vị trí của ký tự. Các ký tự số và chữ sử dụng là 0, 1, 2, 3, ..., 9, A, B, C, D, E, F, trọng số vị trí tính là 16^k khi ký tự ở vị trí thứ k (phải qua trái).

Ví dụ 1 : $Z_{16} = 356_{16}$ sẽ có Z_{10} tương đương là:

$$\begin{aligned}Z_{10} &= 3.16^2 + 5.16^1 + 6.16^0 \\ &= 768 + 80 + 6 = 854_{10}\end{aligned}$$

hoặc

$$\begin{aligned}Z_{16} &= 2A6_{16} \\ Z_{10} &= 2.16^2 + 10.16^1 + 6.16^0 \\ &= 768 + 160 + 6 = 934_{10}\end{aligned}$$

* Khi cần chuyển cách biểu diễn Z_{10} sang Z_{16} tương đương, ta theo quy tắc tương tự là lấy Z_{10} chia liên tiếp cho 16 và đặt phần dư vào số kết quả theo trình tự từ cấp 16^0 trở đi.

Ví dụ 2: Cho $Z_{10} = 442_{10}$ tìm số Z_{16} tương đương.

$$\begin{array}{l} \frac{Z_{10}}{16} = \frac{442}{16} = 27 + \text{dư } 10 \\ \frac{27}{16} = 1 \text{ dư } 11 \\ \frac{1}{16} = 0 + \text{dư } 1 \end{array} \quad \begin{array}{l} \xrightarrow{\text{A}} \\ \xrightarrow{\text{B}} \\ \xrightarrow{\text{Z}_{16} = 1 \text{ B A}} \end{array}$$

Chú ý nếu phép chia $\frac{Z_{10}}{16} = 27,625$ thì phần dư $0,625_{10}$ được nhân với 16 để

có phần dư trong hệ mười sáu: $0,625 \cdot 16 = 10_{10} = Z_{16}$.

* Khi chuyển đổi $Z_{16} \leftrightarrow Z_2$ tương tự như ở hệ tám ta tiến hành biểu diễn theo từng nhóm 4 bit 1 ký tự của hệ mười sáu:

Ví dụ 1: Với $Z_{16} = 9 \text{ E } 3_{16}$ có Z_2 tương đương dạng:

$$Z_2 = 1001 \ 1110 \ 0011_2$$

$$\begin{array}{l} \text{Ví dụ 2:} \quad \text{với } Z_2 = \quad 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \\ \quad \quad \quad = \underbrace{0 \ 0 \ 1 \ 1}_{3} \ \underbrace{1 \ 0 \ 1 \ 1}_{\text{B}} \ \underbrace{1 \ 0 \ 1 \ 0}_{\text{A}} \\ Z_{16} = \quad \quad \quad 3 \quad \quad \text{B} \quad \quad \text{A} \end{array}$$

1.4. CÁC HỆ THỐNG MÃ NHỊ PHÂN THÔNG DỤNG

Khi sử dụng 1 nhóm ký hiệu đặc biệt để biểu diễn một số, một chữ hay một từ nào đó ta nói rằng đã thực hiện việc mã hoá số, chữ hay từ đã có... Nhóm ký hiệu tương ứng được gọi là 1 từ mã (code). Theo ý nghĩa này, các hệ thống số biểu diễn một số đã cho là một hệ thống mã hoá, trong đó mã hoá nhị phân quy ước một số thập phân nào đó có ý nghĩa đặc biệt vì đó là ngôn ngữ thực hiện phân giao tiếp người - máy tính.

1.4.1. Các dạng mã nhị thập phân (BCD)

* Khi biểu diễn 1 ký số trong hệ 10 bằng giá trị nhị phân tương đương của nó ta nhận được mã BCD. Các ký số thập phân từ 0 đến 9 cần có 4 bit nhị phân thể hiện. Nếu sử dụng 10 dòng đầu tiên trong bảng (1.1) đã có ta nhận được mã BCD - tự nhiên hay gọi là mã BCD - 8421 (để nhấn mạnh trọng số của các ký số được sử dụng trong 1 từ mã là $2^3, 2^2, 2^1, 2^0$).

Ví dụ : Cho một số thập phân Z_{10} có dạng:

$$Z_{10} = 9 \quad 6 \quad 5$$

khi đó mã BCD. 8421 của nó có dạng $Z_{BCD} = 1001 \ 0110 \ 0101$

* Chú ý trong bảng 1.1 sáu dòng cuối cùng tương ứng với các số hệ 10 là 10, 11, 12, 13, 14, 15 không được sử dụng (thừa) trong cách biểu diễn mã BCD - 8421.

* Cũng cần nhận xét là mã BCD không phải là một hệ thống số như hệ hai, hệ tám, hệ mười đã nói tới mà chỉ là sự tương ứng từng ký số thập phân với nhóm ký số nhị phân tương đương, do đó cần nhiều bit hơn so với việc biểu diễn mã nhị phân quy ước. Tuy nhiên do có cấu tạo tuân theo trật tự vị trí của hệ thập phân đối với một từ mã 4 bit (vị trí nhóm thứ k từ phải sang trái thể hiện 10^k ở trọng số) nên nó có ưu điểm chính là dễ dàng chuyển đổi $A_{10} \leftrightarrow A_2$ và cấu tạo mạch điện tử sẽ đơn giản hơn (như xét tới ở các chương sau). Nhược điểm quan trọng của mã BCD - 8421 là theo trật tự tự nhiên liên tiếp (trật tự ký số hệ 10) các từ mã có cấu tạo với khoảng cách từ lớn (xác định bằng số bit nhị phân khác trị số nhau của hai từ mã liên tiếp).

Ví dụ: Hai số tự nhiên liên tiếp trong hệ 10: 7 và 8 trong mã BCD - 8421 được biểu diễn là 0111 và 1000, lúc đó cả 4 bit nhị phân tương ứng đều có giá trị thay đổi khi chuyển từ số này sang số kia (khoảng cách từ mã là 4).

1.4.2. Nếu theo một quy luật nào đó, ta chọn 10 trong số 16 tổ hợp trên bảng (1.1) để tạo nên một biểu diễn nhị phân khác của các số tự nhiên từ 0 đến 9, ta sẽ nhận được các dạng mã nhị phân BCD có trọng số khác nhau. Bảng (1.2) đưa ra một số mã BCD thường gặp.

Trong bảng 1.2, một số hệ mười bất kỳ được biểu diễn theo quy luật của mã BCD gồm 4 ký số (4 bit) nhị phân có các trọng số vị trí tương ứng, ví dụ:

$$\begin{aligned} 7_{10} &= 0 \ 1 \ 1 \ 1 = 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \quad \text{với BCD 8421} \\ &= 1 \ 0 \ 0 \ 0 = 1 \cdot 7 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 \quad \text{với BCD 7421} \\ &= 1 \ 0 \ 1 \ 0 = 1 \cdot 5 + 0 \cdot 1 + 1 \cdot 2 + 0 \cdot 1 \quad \text{với BCD 5121} \\ &= 1 \ 1 \ 0 \ 1 = 1 \cdot 2 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 \quad \text{với BCD 2421} \end{aligned}$$

$$\text{hay } 5_{10} = (1 \ 0 \ 0 \ 0)_{BCD \ 5121} = (1 \ 0 \ 1 \ 1)_{BCD \ 2421}$$

Bảng 1.2. CẤU TẠO MÃ BCD VỚI CÁC TRỌNG SỐ KHÁC NHAU

Số hệ mười	Trọng số của mã B C D															
	8	4	2	1	7	4	2	1	5	1	2	1	2	4	2	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	0
5	0	1	0	1	0	1	0	1	1	0	0	0	1	0	1	1
6	0	1	1	0	0	1	1	0	1	0	0	1	1	1	0	0
7	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	1
8	1	0	0	0	1	0	0	1	1	0	1	1	1	1	1	0
9	1	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1

1.4.3. Một vài dạng mã nhị phân khác

1. Mã thừa 3 được xây dựng như sau:

* Lập mã BCD 8421 theo bảng 1.2.

* Cộng thêm 3 đơn vị thập phân hay 0011_2 vào tất cả các từ mã BCD8421 vừa có.

Ví dụ: $4_{10} + 3_{10} = 0100 + 0011$ (BCD 8421)

$= 0111$ mã thừa 3

Vậy ký số 4_{10} trong mã thừa 3 được ký hiệu là 0111, toàn bộ mười từ mã của mã thừa 3 xem trong bảng 1.3, chúng có tính đối xứng trên dưới, từng cặp từ mã là các số bù nhị phân của nhau.

2. Mã vòng Gray có đặc điểm quan trọng là các từ mã liên tiếp nhau luôn chỉ khác nhau một giá trị nhị phân trong 4 giá trị của chúng, khi đó ta có khoảng cách giữa 2 từ mã liên tiếp nhau bằng 1 (do bằng số bit khác trị nhau của chúng) và là một khoảng cách tối thiểu (Việc lập mã vòng Gray dựa trên bìa các nơ hình 1.4c khi chọn 10 ô liên tiếp nhau trong hình này, ô xuất phát có thể chọn bất kỳ và gán cho nó giá trị 0 trong hệ mười, kết thúc ở ô thứ 10 gán cho nó giá trị 9 trong hệ thập phân).

Bảng 1.3. CẤU TẠO CỦA MỘT SỐ MÃ NHỊ PHÂN THÔNG DỤNG

	Số hệ hai				Mã thừa 3				Mã Gray				Mã Gray 3				Mã Johnson				
	B ₃	B ₂	B ₁	B ₀	A ₃	A ₂	A ₁	A ₀	G ₃	G ₂	G ₁	G ₀	G ₃	G ₂	G ₁	G ₀	J ₄	J ₃	J ₂	J ₁	J ₀
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1
2	0	0	1	0	0	1	0	1	0	0	1	1	0	1	1	1	0	0	0	1	1
3	0	0	1	1	0	1	1	0	0	0	1	0	0	1	0	1	0	0	1	1	1
4	0	1	0	0	0	1	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1
5	0	1	0	1	1	0	0	0	0	1	1	1	1	1	0	0	1	1	1	1	1
6	0	1	1	0	1	0	0	1	0	1	0	1	1	1	0	1	1	1	1	1	0
7	0	1	1	1	1	0	1	0	0	1	0	0	1	1	1	1	1	1	1	0	0
8	1	0	0	0	1	0	1	1	1	1	0	0	1	1	1	0	1	1	0	0	0
9	1	0	0	1	1	1	0	0	1	1	0	1	1	0	1	0	1	0	0	0	0

3. Mã Johnson sử dụng 5 bit nhị phân để biểu diễn các số hệ mười. Từ bảng 1.3 với loại mã này ta có nhận xét dãy từ mã liên tiếp ký số 1 dịch dần từ trái qua phải cho đến khi đây ($J_4 = J_3 = J_2 = J_1 = J_0 = 1$) đến lượt ký số 0 dịch dần theo cùng quy luật.

4. Mã Gray - 3 là mã Gray có trật tự từ mã dịch đi 3 hàng về phía dưới (xem hình 1.4c trang 31).

Chú ý : Trong bảng 1.3 chỉ đưa ra một ví dụ về mã vòng Gray là mã Gray 3.

1.5. ĐẠI SỐ LOGIC (ĐẠI SỐ BOOLE)

1.5.1. Đại số logic được hiểu là một tập hợp chỉ gồm các đối tượng có hai trạng thái : có hoặc không có, mệnh đề đúng hoặc sai, các đối tượng này được biểu diễn bằng biến logic. Khi trạng thái của đối tượng là tồn tại (có) ta gán cho biến logic biểu diễn nó giá trị quy ước là 1 và ký hiệu là A, còn khi trạng thái của đối tượng là không tồn tại (không có) ta gán cho biến logic thể hiện nó giá trị quy ước 0 và ký hiệu là $\bar{A}$.

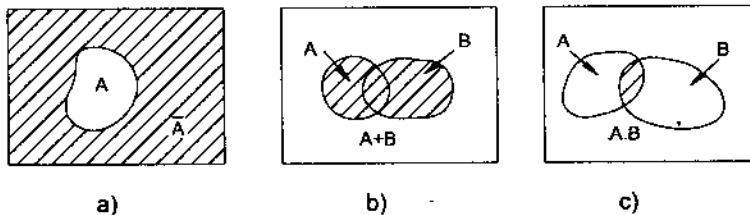
Giữa các biến logic, người ta định nghĩa 3 phép toán cơ sở :

* Phép phủ định logic đối với một biến logic A nào đó (phép đảo) là khi tác động phép toán này, A sẽ nhận giá trị đảo của giá trị ban đầu và ký hiệu là $\bar{A}$.

* Phép cộng logic (phép hoặc) được ký hiệu bằng dấu " + ". Ví dụ: $A + B$ là phép cộng giữa hai biến logic A và B, mỗi biến được gọi là một số hạng và kết quả gọi là một tổng.

* Phép nhân logic (phép và) được ký hiệu bằng dấu " . ". Ví dụ $A.B$ là phép nhân giữa hai biến logic A và B, mỗi biến được gọi là một thừa số của phép nhân, kết quả gọi là tích số. Có thể dùng giản đồ Venn trong lý thuyết tập hợp (xem hình 1.1) để biểu diễn mô tả 3 phép toán logic vừa nêu:

Một trạng thái của đối tượng nào đó luôn luôn có thì biến logic biểu diễn nó luôn ở giá trị 1, còn khi trạng thái của đối tượng luôn luôn không có, giá trị logic của nó luôn là 0. Ta nhận được trong tập hợp này hai hằng số 1 và 0.



Hình 1.1. Đồ thị Venn mô tả 3 phép tính logic cơ bản.

a) Phép phủ định logic; b) Phép cộng logic; c) Phép nhân logic
(phép không- NOT) (phép hoặc - OR) (phép và - AND)

1.5.2. Các tính chất quan trọng của tập hợp các biến logic

1. Khi thực hiện 3 phép toán cơ sở lên các biến logic, ta nhận được một kết quả được gọi là hàm logic. Do tính chất các biến là biến trạng thái nên hàm logic cũng là hàm trạng thái. Khi hàm logic nhận được là do từ nhiều cách tác động tương đương của các phép toán khác nhau ta gọi chúng là tương đương nhau và ký hiệu bằng dấu " = " giữa các kết quả này.

Các tính chất cơ bản:

* Tính hoán vị của phép cộng và phép nhân:

$$A + B = B + A \quad \text{hay} \quad A . B = B . A \quad (1.6)$$

* Tính kết hợp của phép cộng và phép nhân:

$$(A + B) + C = A + (B + C) ; (A \cdot B) \cdot C = A \cdot (B \cdot C) \quad (1.7)$$

* Tính phân phối giữa phép cộng và phép nhân :

$$A (B + C) = A \cdot B + A \cdot C \quad (1.8)$$

* Hai quy tắc của phép phủ định:

$$\overline{(\overline{A})} = A \quad ; \quad \overline{(\overline{A})} = A \quad (1.9)$$

* Bốn quy tắc của phép cộng:

$$\begin{aligned} A + A &= A & A + \overline{A} &= 1 \\ A + 0 &= A & A + 1 &= 1 \end{aligned} \quad (1.10)$$

* Bốn quy tắc của phép nhân:

$$\begin{aligned} A \cdot A &= A & A \cdot \overline{A} &= 0 \\ A \cdot 1 &= A & A \cdot 0 &= 0 \end{aligned} \quad (1.11)$$

* Tính chất hấp thụ:

$$A \cdot (A + B) = A \quad (1.12)$$

* Tính nhất quán : nếu $A + B = B$

$$\text{thì } A \cdot B = A$$

(1.13)

* Luật De Morgan lập hàm phủ định của một hàm:

$$\begin{aligned} \overline{A + B} &= \overline{A} \cdot \overline{B} \\ \overline{A \cdot B} &= \overline{A} + \overline{B} \end{aligned} \quad (1.14)$$

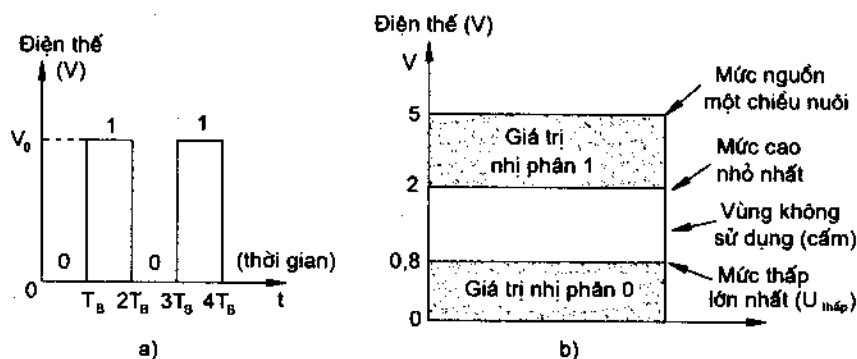
$$* \quad A + \overline{A} \cdot B = A + B \quad (1.15)$$

Các hệ thức từ (1.6) đến (1.15) có thể dễ dàng chứng minh tính đúng đắn của chúng khi ta sử dụng đồ thị Venn hoặc sử dụng các công tắc trạng thái A, B trong một mạch điện với phép cộng là mắc song song các công tắc, phép nhân là mắc nối tiếp chúng, trạng thái nối mạch có giá trị 1, ngắt mạch có giá trị 0.

1.5.3. Phương pháp dùng mức điện thế biểu diễn giá trị của biến và hàm logic

* Để biểu diễn giá trị logic của một biến logic hay hàm logic người ta thường sử dụng một xung điện áp (một tín hiệu số) như thể hiện trên hình 1.2.

* Khi cần biểu diễn một dãy ký số nhị phân ta dùng một dãy xung tuần hoàn như hình 1.2a biểu thị số nhị phân 0 1 0 1, chu kỳ xung vuông góc chính là chu kỳ 1 bit (gọi là chu kỳ bit: T_B).



Hình 1.2. Biểu diễn giá trị biến logic bằng mức điện thế tương ứng

* Để xác định trạng thái điện thế cao hay trạng thái điện thế thấp của một xung vuông góc xuất hiện (đại diện cho giá trị một bit nhị phân là 1 hay là 0) cần đưa ra một tiêu chuẩn đánh giá: từ mức nào trở lên được quy định là logic 1 hay từ mức nào trở xuống tới 0V được quy định là logic 0. Trên hình 1.2b, ví dụ theo quy định ta có:

Khi $0 \leq V \leq 0,8V$ ta có mức V biểu diễn trị nhị phân 0.

Khi $2V \leq V \leq 5V$ ta có mức V biểu diễn trị nhị phân 1.

Nếu trạng thái giá trị của V đều có giá trị dương, ta gọi đây là logic dương. Ngược lại với 1 xung điện thế âm ta có logic âm.

* Vùng giá trị $0,8V < V < 2V$ không được sử dụng, ta gọi đây là vùng cấm vì khi đó giá trị của biến logic sẽ không xác định.

* Khi các xung điện thế xuất hiện đồng thời ở các cổng ra và các cổng vào của các mạch điện tử số nào đó ta nhận được phương thức truyền song song các ký số nhị phân. Trong trường hợp khác, tại 1 cổng vào của mạch số xuất hiện lần lượt các xung sau từng chu kỳ bit T_B có trật tự đến trước vào trước, đến sau vào sau gọi là phương thức truyền nối tiếp các ký số nhị phân (các bit).

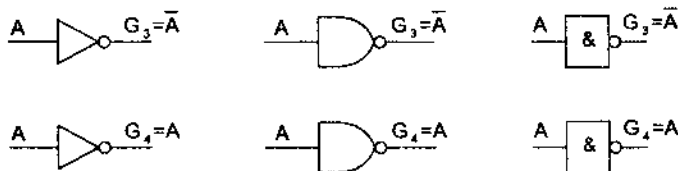
1.5.4. Các hàm logic sơ cấp

1. Nhóm các hàm 1 biến $G(A)$ gồm 4 hàm cơ sở

$G_1 = 0$ (A luôn bằng 0) $F_0 \equiv G_3 = \bar{A}$ (hàm bù của A – còn gọi là hàm đảo NOT)

$G_2 = 1$ (A luôn bằng 1) $G_4 = A$ (hàm lặp của A – YES)

ký hiệu quy ước của G_3 và G_4 cho trên hình 1.3.



Hình 1.3. Ký hiệu quy ước hàm NOT và YES

2. Nhóm các hàm 2 biến $F(A, B)$ cho trên bảng 1.4

Bảng 1.4. CÁC HÀM HAI BIẾN QUAN TRỌNG NHẤT

Biến Hàm	A	B	0	1	0	1	Biểu thức đại số	Tên gọi tiếng Việt	Tên gọi quốc tế	Ký hiệu quy ước (h1.6)
F_1	0	0	0	1	0	1	$F_1 = A.B$	Nhân logic (và)	AND	
F_2	0	1	1	1	0	1	$F_2 = A+B$	Cộng logic (hoặc)	OR	
F_3	1	1	1	0	0	1	$F_3 = \overline{A.B} = A \vee B$	Nhân đảo	NAND	
F_4	1	0	0	0	0	1	$F_4 = \overline{A+B} = A \downarrow B$	Cộng đảo	NOR (PIERCE)	
F_5	0	0	1	0	0	1	$F_5 = A.\bar{B} = A \supset B$	Cấm B	INHIBITION	
F_6	0	1	0	0	0	1	$F_6 = \bar{A}.B = A \supset B$	Cấm A	INHIBITION	
F_7	0	1	1	0	0	1	$F_7 = AB + \bar{A}\bar{B} = A \oplus B$	Khác dấu	EXCLUSIVE XOR	
F_8	1	0	0	1	0	1	$F_8 = A.B + \bar{A}\bar{B} = A \vee B$	Đồng dấu	EXCLUSIVE NOR (XNOR)	
F_9	1	0	1	1	0	1	$F_9 = A + \bar{B} = B \rightarrow A$	Kéo theo A	IMPLICATION	
F_{10}	1	1	0	1	0	1	$F_{10} = \bar{A} + B = A \rightarrow B$	Kéo theo B	IMPLICATION	

1.5.5. Hệ hàm logic đầy đủ

Từ một tổ hợp các hàm logic sơ cấp nào đó ta có thể xây dựng được một hàm logic bất kỳ - một nhóm các hàm sơ cấp mà từ đó có thể xây dựng được các hàm logic khác được gọi là một hệ hàm đầy đủ.

Có 4 hệ hàm đầy đủ

- a) Hệ bao gồm các hàm $F_0 = \bar{A} (= G_3)$; $F_1 = A \cdot B$ và $F_2 = A + B$
- b) Hệ chỉ dùng hàm $F_3 = \overline{A \cdot B}$ (NAND)
- c) Hệ chỉ dùng hàm $F_4 = \overline{A + B}$ (NOR)
- d) Hệ gồm 2 hàm $F_7 = A \oplus B$ và $F_5 = A \cdot \bar{B}$ (hoặc $F_6 = \bar{A} \cdot B$)

1.6. PHƯƠNG PHÁP BIỂU DIỄN HÀM LOGIC VÀ TỐI THIỂU HÀM LOGIC

1.6.1. Biểu diễn hàm logic bằng bảng chân lý

Hàm logic có thể biểu diễn ở dạng một bảng liệt kê các giá trị có thể của biến và giá trị tương ứng của hàm gọi là bảng chân lý (hay bảng trạng thái) giống như bảng 1.5. Như vậy với hàm 2 biến ta có bảng gồm 3 cột và 4 dòng, với hàm 3 biến ta có bảng chân lý gồm 4 cột và $2^3 = 8$ dòng (tương ứng với mọi trạng thái tổ hợp biến có thể có) và với hàm 4 biến ta sẽ có 5 cột và $2^4 = 16$ dòng...

Biến		Hàm
A	B	F_6
0	0	1
0	1	0
1	0	0
1	1	1

$$F_6 = \bar{A} \cdot \bar{B} + A \cdot B$$

Biến		Hàm
A	B	F_7
0	0	0
0	1	1
1	0	1
1	1	0

$$F_7 = \bar{A} \cdot B + A \cdot \bar{B}$$

Biến			Hàm
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$F = ABC + \bar{A} \bar{B} C + A \bar{B} \bar{C} + A B \bar{C}$$

1.6.2. Biểu diễn hàm logic bằng biểu thức

Khái niệm về MAXTERM (Mactec) và MINTERM (Mintec). Phương pháp biểu diễn hàm logic bằng biểu thức giải tích có hai dạng cơ bản:

* Dạng tổng các tích các biến, mỗi số hạng của tổng được gọi là 1 mintec (đủ biến) ký hiệu là m_i .

* Dạng tích các tổng các biến, mỗi thừa số của tích được gọi là 1 mactec (đủ biến) ký hiệu là M_i (chỉ số i tính trong hệ mười).

1. Bảng các m_i và M_i của hàm 2 biến $F(A, B)$, hàm 3 biến $F(A, B, C)$ và hàm 4 biến $F(A, B, C, D)$

Bảng 1.5. a) CÁC m_i VÀ M_i CỦA HÀM 2 BIẾN ($k=2$) ;

b) CÁC m_i VÀ M_i CỦA HÀM 3 BIẾN ($k=3$)

Biến		Mintec	Mactec
A	B	m_i	M_i
0	0	$\bar{A}\bar{B}=m_0$	$\bar{A}+\bar{B}=M_3$
0	1	$\bar{A}B=m_1$	$\bar{A}+B=M_2$
1	0	$A\bar{B}=m_2$	$A+\bar{B}=M_1$
1	1	$AB=m_3$	$A+B=M_0$

a)

Biến			m_i	M_i
A	B	C		
0	0	0	$\bar{A}\bar{B}\bar{C}=m_0$	$\bar{A}+\bar{B}+\bar{C}=M_7$
0	0	1	$\bar{A}\bar{B}C=m_1$	$\bar{A}+\bar{B}+C=M_6$
0	1	0	$\bar{A}B\bar{C}=m_2$	$\bar{A}+B+\bar{C}=M_5$
0	1	1	$\bar{A}BC=m_3$	$\bar{A}+B+C=M_4$
1	0	0	$A\bar{B}\bar{C}=m_4$	$A+\bar{B}+\bar{C}=M_3$
1	0	1	$A\bar{B}C=m_5$	$A+\bar{B}+C=M_2$
1	1	0	$AB\bar{C}=m_6$	$A+B+\bar{C}=M_1$
1	1	1	$ABC=m_7$	$A+B+C=M_0$

b)

Ta cần chú ý trong bảng 1.5 khi biến có giá trị bù (trị 0) ta ký hiệu là $\bar{A}$, còn khi biến ở dạng trực tiếp (nhận giá trị 1) ta ký hiệu tương ứng là A. Trong cùng một hàng của bảng 1.5a,b hay c; tổng chỉ số m_i và M_j này luôn bằng (2^k-1) trong đó k là số biến của hàm, cặp m_i và M_j này ($i+j = 2^k-1$) được gọi là cùng tên nhau, ví dụ trong bảng 1.5b cặp m_4 và M_3 hay cặp m_6 và M_1 .

c) BẢNG CÁC m_i VÀ M_i CỦA HÀM 4 BIẾN ($k = 4$)

Biến				Minterc m_i	Maxtec M_i
A	B	C	D		
0	0	0	0	$m_0 = \bar{A}\bar{B}\bar{C}\bar{D}$	$M_{15} = \bar{A} + \bar{B} + \bar{C} + \bar{D}$
0	0	0	1	$m_1 = \bar{A}\bar{B}\bar{C}D$	$M_{14} = \bar{A} + \bar{B} + \bar{C} + D$
0	0	1	0	$m_2 = \bar{A}\bar{B}C\bar{D}$	$M_{13} = \bar{A} + \bar{B} + C + \bar{D}$
0	0	1	1	$m_3 = \bar{A}\bar{B}CD$	$M_{12} = \bar{A} + \bar{B} + C + D$
0	1	0	0	$m_4 = \bar{A}B\bar{C}\bar{D}$	$M_{11} = \bar{A} + B + \bar{C} + \bar{D}$
0	1	0	1	$m_5 = \bar{A}B\bar{C}D$	$M_{10} = \bar{A} + B + \bar{C} + D$
0	1	1	0	$m_6 = \bar{A}BC\bar{D}$	$M_9 = \bar{A} + B + C + \bar{D}$
0	1	1	1	$m_7 = \bar{A}BCD$	$M_8 = \bar{A} + B + C + D$
1	0	0	0	$m_8 = A\bar{B}\bar{C}\bar{D}$	$M_7 = A + \bar{B} + \bar{C} + \bar{D}$
1	0	0	1	$m_9 = A\bar{B}\bar{C}D$	$M_6 = A + \bar{B} + \bar{C} + D$
1	0	1	0	$m_{10} = A\bar{B}C\bar{D}$	$M_5 = A + \bar{B} + C + \bar{D}$
1	0	1	1	$m_{11} = A\bar{B}CD$	$M_4 = A + \bar{B} + C + D$
1	1	0	0	$m_{12} = AB\bar{C}\bar{D}$	$M_3 = A + B + \bar{C} + \bar{D}$
1	1	0	1	$m_{13} = AB\bar{C}D$	$M_2 = A + B + \bar{C} + D$
1	1	1	0	$m_{14} = ABC\bar{D}$	$M_1 = A + B + C + \bar{D}$
1	1	1	1	$m_{15} = ABCD$	$M_0 = A + B + C + D$

Từ bảng 1.5 ta có thể biểu diễn giải tích 1 hàm logic 2, 3 hay 4 biến bất kỳ bằng cách viết ở dạng hệ thức (1.16) và (1.17) như sau:

$$F(A, B, C) = \sum_i m_i \quad (1.16)$$

$$F(A, B, C, \dots) = \prod_i M_i \quad (1.17)$$

để làm rõ hơn ta xét ví dụ sau:

Cho các hàm

$$F_7 = A\bar{B} + \bar{A}B$$

$$F = \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

khi đó có thể viết F_7 theo bảng 1.5a ở dạng:

$$F_7 = m_1 + m_2 = \sum m(1, 2)$$

Tương tự theo bảng 1.5b ta có :

$$F = m_1 + m_4 + m_5 + m_6 + m_7 = \Sigma m(1, 4, 5, 6, 7)$$

Áp dụng luật De Morgan theo công thức (1.14) ta có :

$$\overline{F_7} = \overline{AB + AB} = \overline{AB} \cdot \overline{AB} = (\overline{A} + \overline{B}) \cdot (A + B)$$

$$\begin{aligned} \overline{F} &= \overline{ABC} \cdot \overline{ABC} \cdot \overline{ABC} \cdot \overline{ABC} \cdot \overline{ABC} \\ &= (A + B + \overline{C})(\overline{A} + B + C)(\overline{A} + B + \overline{C})(\overline{A} + \overline{B} + C)(\overline{A} + \overline{B} + \overline{C}) \end{aligned}$$

đối chiếu với bảng 1.5 ta nhận được :

$$\overline{F_7} = M_1, M_2 = \Pi M(1, 2)$$

$$\overline{F} = M_1 \cdot M_4 \cdot M_5 \cdot M_6 \cdot M_7 = \Pi M(1, 4, 5, 6, 7)$$

Chú ý các hàm $\overline{F_7}$ và $\overline{F}$ thu được trong ví dụ trên là các hàm bù (phủ định) của hai hàm đã cho. Biểu diễn cho F_7 và F (dạng tích các tổng các biến) được thiết lập từ kết quả (1.16) của dạng tổng các mintec bằng cách thiết lập các mactec M_j ; với các chỉ số $j \neq i$ trong bảng 1.5, khi áp dụng điều nhận xét trên ta có:

$$F_7 = M_0 \cdot M_3 = \Pi M(0, 3)$$

$$F = M_0 \cdot M_2 \cdot M_3 = \Pi M(0, 2, 3)$$

2. Các tính chất quan trọng của m_i và M_j với hàm logic k biến

$$* \text{ Khi } i + j = 2^k - 1 \text{ thì } \overline{m_i} = M_j \quad (1.18)$$

$$\text{ví dụ trong bảng 1.5b : } \overline{m_i} = M_6$$

$$* \Sigma m(0, 1, 2, 3) = 1 \text{ (với hàm 2 biến)}$$

$$\Pi M(0, 1, 2, 3) = 0$$

$$\text{hay với hàm logic k biến: } \sum_{i=0}^{2^k-1} m_i = 1$$

$$\prod_{j=0}^{2^k-1} M_j = 0 \quad (1.19)$$

$$* \text{ Với hàm logic k biến : } m_i \cdot m_j = 0 \text{ nếu } i \neq j$$

$$M_i + M_j = 1 \text{ nếu } i \neq j \quad (1.20)$$

1.6.3. Biểu diễn hàm logic bằng phương pháp hình học (bìa các nô)

Hàm logic k biến được biểu diễn thành 1 bảng có 2^k các ô vuông (mỗi ô tương ứng với một mintec m_i của hàm). Các tổ hợp biến phải xếp theo thứ tự của mã Gray (bảng 1.3) nghĩa là 2 ô (2 mintec) kế nhau chỉ được phép có 1 biến khác trị số. Hình 1.4 đưa ra bìa các nô của các hàm logic từ 2 tới 5 biến. Ta thấy ô nào ứng với giá trị mintec $m_i = 1$ thì ta gán giá trị 1 vào nó, còn ô nào có trị $m_i = 0$ thì ta bỏ trống. Khi đó ta được biểu diễn bìa các nô của một hàm logic nào đó đã cho trước như các ví dụ trên hình 1.5 tương ứng. Cần lưu ý bên mép hàng và cột ta ghi các trị số giá trị của biến và ký hiệu biến tương ứng theo đúng trật tự quy định để tránh nhầm lẫn.

A \ B	0	1
0	m_0 $\bar{A}\bar{B}$	m_2 $A\bar{B}$
1	m_1 $\bar{A}B$	m_3 AB

a)

AB \ C	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$ m_0	$\bar{A}B\bar{C}$ m_2	$A\bar{B}\bar{C}$ m_4	$AB\bar{C}$ m_6
1	$\bar{A}\bar{B}C$ m_1	$\bar{A}BC$ m_3	$AB\bar{C}$ m_7	ABC m_5

b)

AB \ CD	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$ m_0	$\bar{A}B\bar{C}\bar{D}$ m_4	$A\bar{B}\bar{C}\bar{D}$ m_{12}	$AB\bar{C}\bar{D}$ m_8
01	$\bar{A}\bar{B}C\bar{D}$ m_1	$\bar{A}BC\bar{D}$ m_5	$A\bar{B}C\bar{D}$ m_{13}	$AB\bar{C}\bar{D}$ m_9
11	$\bar{A}\bar{B}CD$ m_3	$\bar{A}BCD$ m_7	$A\bar{B}CD$ m_{15}	$AB\bar{C}D$ m_{11}
10	$\bar{A}B\bar{C}\bar{D}$ m_2	$\bar{A}BC\bar{D}$ m_6	$A\bar{B}C\bar{D}$ m_{14}	$AB\bar{C}\bar{D}$ m_{10}

c)

ABC \ DE	000	001	011	010	110	111	101	100
00	$\bar{A}\bar{B}\bar{C}\bar{D}\bar{E}$ m_0	$\bar{A}\bar{B}\bar{C}D\bar{E}$ m_4	$\bar{A}\bar{B}C\bar{D}\bar{E}$ m_{12}	$\bar{A}\bar{B}CD\bar{E}$ m_8	$\bar{A}B\bar{C}\bar{D}\bar{E}$ m_{24}	$\bar{A}B\bar{C}D\bar{E}$ m_{20}	$\bar{A}BC\bar{D}\bar{E}$ m_{28}	$\bar{A}BCD\bar{E}$ m_{24}
01	$\bar{A}\bar{B}\bar{C}\bar{D}E$ m_1	$\bar{A}\bar{B}\bar{C}DE$ m_5	$\bar{A}\bar{B}C\bar{D}E$ m_{13}	$\bar{A}\bar{B}CDE$ m_9	$\bar{A}B\bar{C}\bar{D}E$ m_{25}	$\bar{A}B\bar{C}DE$ m_{21}	$\bar{A}BC\bar{D}E$ m_{29}	$\bar{A}BCDE$ m_{25}
11	$\bar{A}\bar{B}C\bar{D}\bar{E}$ m_3	$\bar{A}\bar{B}CDE$ m_7	$\bar{A}B\bar{C}\bar{D}\bar{E}$ m_{15}	$\bar{A}B\bar{C}D\bar{E}$ m_{11}	$\bar{A}BC\bar{D}\bar{E}$ m_{27}	$\bar{A}BCD\bar{E}$ m_{23}	$\bar{A}BC\bar{D}E$ m_{31}	$\bar{A}BCDE$ m_{27}
10	$\bar{A}B\bar{C}\bar{D}\bar{E}$ m_2	$\bar{A}B\bar{C}D\bar{E}$ m_6	$\bar{A}BC\bar{D}\bar{E}$ m_{14}	$\bar{A}BCD\bar{E}$ m_{10}	$A\bar{B}\bar{C}\bar{D}\bar{E}$ m_{26}	$A\bar{B}\bar{C}D\bar{E}$ m_{30}	$A\bar{B}C\bar{D}\bar{E}$ m_{22}	$A\bar{B}CD\bar{E}$ m_{18}

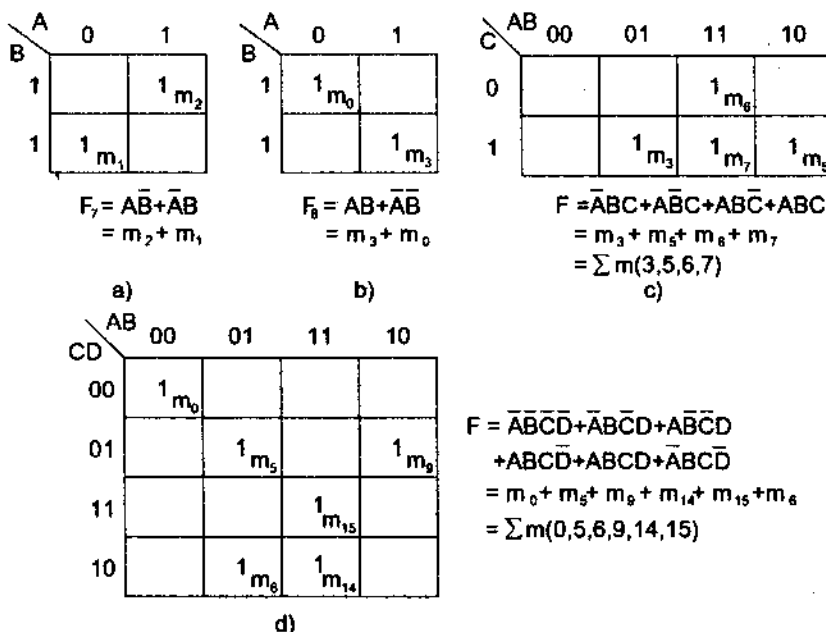
d)

Hình 1.4. Bìa các nô của hàm logic
a) 2 biến ; b) 3 biến ; c) 4 biến ; d) 5 biến.

Như vậy khi lập bìa các nô cho 1 hàm logic nào đó ta cần thực hiện các bước:

* Lập bìa các nô ứng với số biến của hàm đã cho, chú ý hai ô kế nhau trong bìa phải có khoảng cách từ mã nhị phân là tối thiểu (khác nhau chỉ có 1 giá trị nhị phân).

* Sau khi đã đủ các ô trống (đúng quy tắc) các mintec m_i có mặt trong biểu thức của hàm sẽ được điền 1 vào vị trí của ô tương ứng trong bìa, nghĩa là trong biểu thức (1.14) của hàm có bao nhiêu số hạng m_i sẽ có đủ bấy nhiêu ô có trị 1 trong bìa các nô.



Hình 1.5. Cách biểu diễn hàm logic bằng bìa các nô

1.6.4. Rút gọn (tối thiểu hoá) hàm logic

Bài toán kỹ thuật liên quan tới hàm logic rất đa dạng, ta cần quan tâm tới vấn đề làm cách nào để dễ dàng thực hiện (giải) bài toán nhờ các mạch điện tử chỉ gồm các phần tử logic sơ cấp, qua đó chi phí kinh tế kỹ thuật đạt tới tối ưu, có độ tin cậy và độ chuẩn hoá cao, sẵn có trên thị trường. Vì vậy ta cần phải rút gọn hàm logic.

1. Phương pháp tối thiểu bằng giải tích dựa trên các tính chất của đại số logic, các hệ thức đã biết (1.8) đến (1.13) khi số biến logic không nhiều biến đổi trực tiếp biểu thức giải tích của hàm.

Ví dụ 1 : Rút gọn hàm sau:

$$\begin{aligned} F(A, B, C, D) &= A\bar{B} + C + \bar{A}\bar{C}D + \bar{B}CD \\ &= A\bar{B} + C + \bar{C}(\bar{A}D + BD) \end{aligned}$$

Áp dụng tính chất $1.C = C$ và $A + \bar{A}B = A + B$ có:

$$\begin{aligned} F(A, B, C, D) &= A\bar{B} + C.1 + \bar{C}(\bar{A}D + BD) \\ A\bar{B} + C + (\bar{A}D + BD) &= A\bar{B} + C.1 + D(\bar{A} + B) \end{aligned}$$

dùng tính chất (1.13) $(\bar{A} + B) = \overline{A \cdot \bar{B}}$ và tính chất $X + \bar{X}Y = X + Y$

$$F = A\bar{B} + C + D\bar{A\bar{B}} = C + D + A\bar{B}$$

Ví dụ 2: Rút gọn hàm $F(A, B, C) = ABC + \bar{A}BC + A\bar{B}C + AB\bar{C}$

Áp dụng tính chất $X + X = X$ (hệ thức 1.10) cho ABC có

$$ABC + \bar{A}BC + ABC + A\bar{B}C + ABC + AB\bar{C} = F$$

nhóm từng 2 số hạng, chú ý tính chất (1.11) và (1.10) có

$$\begin{aligned} F &= (A + \bar{A})BC + (B + \bar{B})AC + (C + \bar{C})AB \\ F &\doteq BC + AC + AB \end{aligned}$$

2. Phương pháp tối thiểu bằng bìa các nô sử dụng quy tắc dán ô kề nhau (quy tắc các nô)

"Nếu có $2, 4, 8 \dots 2^n$ ô có trị 1 nằm kề nhau, ta có thể dán chúng lại thành 1 ô lớn đại diện cho 1 số hạng rút gọn đi 1, 2, 3 ... n biến". Khi sử dụng quy tắc các nô cần lưu ý mấy trường hợp sau:

- Số các ô dán lại phải bằng 2^n (n là số nguyên 0, 1, 2, 3...) hợp thành một khối vuông hay chữ nhật. Sau khi kết thúc việc dán ô, số lượng ô trị "1" trong một khối là tối đa có thể và số lượng khối độc lập là tối thiểu.

- Hai hay nhiều ô nằm ở 2 mép của bìa tính theo hàng hay theo cột cũng được coi là kề nhau.

- Một hoặc vài ô có trị 1 có thể tham gia dán nhiều lần vào các nhóm khác nhau (nhóm độc lập, không chứa nhau).

- Không được thực hiện dán các ô mà sau khi dán, ô lớn có được lại chứa nhau hay chứa tất cả các ô con đã được dán từ trước đó.

- Khi chuyển từ ô này sang ô kia trong một khối đã nhóm, biến logic nào bị đảo trị số sẽ bị loại bỏ trong biểu diễn số hạng đã thu gọn theo hệ quả của quy tắc (1-10) đã nêu ở trang 24.

Để làm rõ quy tắc ta nêu vài ví dụ minh họa.

Ví dụ 1: Hãy rút gọn hàm $F(A,B,C) = ABC + \bar{A}BC + A\bar{B}C + ABC$

Hàm F có bìa các nô cho trên hình 1.5e gồm 4 ô có trị 1 ứng với các mintec m_3, m_5, m_6 và m_7 . Thực hiện dán m_3 với m_7 , m_6 với m_7 và m_5 với m_7 ta được 3 ô mới ký hiệu tương ứng là X_1, X_2 và X_3 , các ô này có giá trị :

$$X_1 = m_3 + m_7 = BC$$

$$X_2 = m_6 + m_7 = AB$$

$$X_3 = m_5 + m_7 = AC$$

kết quả ta được hàm F đã rút gọn:

$$\begin{aligned} F &= X_1 + X_2 + X_3 \\ &= BC + AB + AC \end{aligned}$$

C \ AB	00	01	11	10
			1 m_7	
0				
1		1 m_3	1 m_5	1 m_6

Hình 1.5e

Ví dụ 2 : Với hàm F cho trên hình 1.5d ta có thể dán các ô m_6 với m_{14} cho kết quả là $BC\bar{D}$ và các ô m_{14} với m_{15} cho kết quả là ABC , vậy sau khi rút gọn hàm hình 1.5d có dạng:

$$\begin{aligned} F(A, B, C, D) &= \sum m(0, 5, 6, 9, 14, 15) \\ &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}BC\bar{D} + BCD + A\bar{B}CD + ABC \end{aligned}$$

Ví dụ 3 : Rút gọn hàm $G(A, B, C, D)$ cho trên bìa các nô (hình 1.6). Biểu thức đầy đủ của hàm G có dạng :

$$G = \sum m(0, 1, 2, 4, 6, 7, 8, 9, 10, 11, 12, 14)$$

gồm 12 số hạng có đủ mặt các biến ABCD. Ta có thể thực hiện dán ô như sau:

G \ AB	00	01	11	10
00	1 m_0	1 m_4	1 m_{12}	1 m_8
01	1 m_1	1 m_5	1 m_{13}	1 m_9
11	1 m_3	1 m_7	1 m_{15}	1 m_{11}
10	1 m_2	1 m_6	1 m_{14}	1 m_{10}

Hình 1.6

Ô lớn X_1 gồm có $m_0 + m_4 + m_{12} + m_8 + m_2 + m_6 + m_{14} + m_{10}$

kết quả là $X_1 = \bar{D}$.

$$\text{Ô lớn } X_2 = m_0 + m_1 + m_8 + m_9 = \overline{B}C$$

$$\text{Ô lớn } X_3 = m_8 + m_9 + m_{10} + m_{11} = A\overline{B}$$

$$\text{Ô lớn } X_4 = m_6 + m_7 = \overline{A}BC$$

Hàm G sau khi rút gọn có dạng:

$$G = X_1 + X_2 + X_3 + X_4 = \overline{D} + \overline{B}C + A\overline{B} + \overline{A}BC$$

Trong các ví dụ trên ta lưu ý rằng trong 1 ô lớn sau khi đã dán các ô nhỏ, các biến logic nào có giá trị thay đổi thì sẽ không còn có mặt trong biểu thức thu gọn nữa và có những ô nhỏ trong ví dụ 2 (m_0, m_5, m_9) không thể dán với một ô nào khác vì chúng không có ô trị 1 nào khác nằm kề bên về mọi phía. Ta cũng cần chú ý rằng hàm logic còn có một cách biểu diễn bằng các sơ đồ logic thực hiện nó, việc này sẽ được xét tới ở chương tiếp sau, khi đã đề cập tới các cổng logic cơ bản.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 1

1.1. Hãy phân biệt đại lượng tương tự và đại lượng số.

Phân loại các đại lượng sau là tương tự hay số?

- Dòng điện chảy qua bóng đèn thấp sáng.
- Độ cao của một máy bay.
- Nhiệt độ trong phòng.
- Lượng hạt cát trên bãi biển.
- Các công tắc điện trong một phòng ở.
- Lượng nước trong 1 bể chứa nước.

1.2. Các hệ thống số đếm dùng để làm gì?

Hãy phân biệt các hệ đếm hai, hệ đếm mười, hệ đếm tám và hệ đếm mười sáu?

Tại sao lại nói hệ thống số đếm nhị phân (hệ hai) và hệ thống số đếm thập phân (hệ mười) là hai hệ thống số quan trọng nhất?

1.3. Hãy tự viết lại các chữ số đầu tiên của 4 hệ thống số đếm quan trọng nhất được sử dụng trong kỹ thuật số và học thuộc bảng đối chiếu tương đương cách biểu diễn một con số trong bốn hệ thống số nói trên.

1.4. Đổi các số sau trong hệ đếm mười sang hệ đếm hai bằng hai phương pháp:

a) $A_{10} = 85$

b) $A_{10} = 164$

c) $A_{10} = 1890$

$A_{10} = 73$

$A_{10} = 200$

$A_{10} = 1945$

d) Cần bao nhiêu bit nhị phân để biểu diễn số thập phân 1 triệu?

1.5. Đổi các số sau trong hệ đếm hai sang hệ đếm mười

a) 1 0 0 1 0 1

c) 1 1 0 1 0 1 1 0

b) 1 1 1 1 0 0

d) 1 0 0 1 0 0 1 1

1.6. Hãy đổi các số hệ mười sáu sau sang hệ mười và sang hệ hai BCD 8421

a) 5 F

b) 7 A

c) 8 B

d) 9 C

e) 4 D

f) 6 E

1.7. Hãy đổi các số hệ tám sang hệ mười và sang hệ hai tự nhiên

a) 47_8

b) 56_8

c) 124_8

d) 31_8

e) 64_8

f) 372_8

1.8. Hãy viết 4 số kế tiếp trong các dãy số sau:

a) 11001_2

11010_2

$1101_2 \dots \dots \dots$

b) 624_8

625_8

$626_8 \dots \dots \dots$

c) $9D_{16}$

$9E_{16}$

$9F_{16} \dots \dots \dots$

1.9. Hãy chuyển đổi các số sau sang biểu diễn tương đương khác:

a) $Z_{16} = 24AE_{16} \rightarrow A_{10} ? \rightarrow A_2 ?$

b) $Z_{16} = A6F2_{16} \rightarrow A_{10} ? \rightarrow A_2 ?$

c) $A_{10} = 3118_{10} \rightarrow Z_{16} ? \rightarrow A_2 ?$

d) $A_{10} = 9785_{10} \rightarrow Z_{16} ? \rightarrow A_2 ?$

e) Hãy viết các số kế tiếp sau dãy A9C, A9D, A9E, ..., ..., ..., ... ?

f) Tương tự câu hỏi (e) với 5 số kế tiếp của dãy sau:

1AFB, 1AFC, 1AFD, ..., ..., ..., ... ?

1.10. Hãy tính tổng các số nhị phân sau, thử lại với các số hệ mười.

a) $1010 + 0111 + 1100$

b) $11010 + 0101 + 101$

c) $10010110 + 01110101$

1.11. Hãy tính hiệu hai số nhị phân, kiểm tra lại kết quả ở biểu diễn hệ mười:

a) $110101 - 100101$

b) $10010110 - 100110$

1.12. Tìm tích toàn phần hai số nhị phân:

a) 101101×101

b) 11001×1011

c) Đổi các thừa số và kết quả ra số hệ mười để kiểm tra lại phép tính đã thực hiện.

1.13. a) Hãy viết lại và học thuộc cách biểu diễn 10 số tự nhiên đầu tiên trong hệ mười thành mã BCD 8421, mã BCD 7421 và mã BCD 2421.

b) Nêu cách xây dựng mã thừa 3, mã vòng Gray và mã Johnson. Hãy viết lại bảng mã thừa 3, mã Gray và mã Johnson cho 10 số tự nhiên đầu tiên của hệ mười.

c) Hãy chứng minh rằng $G_i = B_i \oplus B_{i+1}$ để tạo mã Gray trong bảng (1.3).

1.14. Nêu các tính chất cơ bản của đại số logic.

Hãy vẽ đồ thị Venn mô tả các phép tính $X_1 + X_2 + X_3$, $X_1 X_2 X_3$ và $\bar{X}$ với X_i là các đối tượng có thể mô tả được bằng một hàm logic.

1.15. Hãy viết nhóm các hàm logic 2 biến sơ cấp và ghi nhớ:

a) Biểu thức hàm

b) Tên gọi của hàm

c) Bảng chân lý (bảng trạng thái) của chúng

d) Ký hiệu quy ước các hàm trên.

1.16. Hãy trình bày tóm tắt 3 cách biểu diễn hàm logic :

a) Biểu diễn bằng bảng chân lý

b) Biểu diễn bằng biểu thức giải tích (của các mintec và mactec)

c) Biểu diễn bằng bìa các nô.

1.17. Cho các hàm 3 biến :

$$X_1 = ABC + \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}B\bar{C}$$

$$X_2 = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

a) Viết bảng chân lý và bìa các nô của X_1 và X_2

b) Tìm mối liên hệ logic nếu có của X_1 với X_2

c) Xác định hàm $X_1 X_2 + \bar{X}_1 \bar{X}_2 = Z$

1.18. Với hàm logic 3 biến, hãy chứng minh $F_1 = F_2$ và $F_3 = F_4$ (bằng cách thiết lập bảng chân lý)

với $F_1 = A + B.C$

$$F_2 = (A + B)(A + C)$$

$$F_3 = AB + \bar{A}C + \bar{B}C$$

$$F_4 = AB + \bar{A}C$$

1.19. Cho hàm logic 4 biến :

$$G(A, B, C, D) = \sum m(0, 3, 5, 6, 9, 10, 12, 15)$$

a) Hãy vẽ bìa các nô cho G.

b) Thiết lập biểu thức của $\bar{G} = \sum m_i$ (gồm 8 số hạng còn lại trên bìa các nô của G).

1.20. Tại sao phải thực hiện rút gọn hàm logic ?

Có mấy cách thực hiện việc rút gọn này ?

Hãy trình bày cụ thể các quy tắc rút gọn theo phương pháp các nô.

1.21. Cho các hàm 4 biến F_1 và F_2 , G_1 và G_2

$$F_1 = A + BCD$$

$$F_2 = (A + B)(A + C)(A + D)$$

$$G_1 = A(B + C + D)$$

$$G_2 = AB + AC + AD$$

Hãy chứng minh rằng $F_1 = F_2$ và $G_1 = G_2$

1.22. Cho hàm 4 biến $F(A, B, C, D) = \overline{(A \oplus B)}(C + D)$

a) Hãy đưa hàm F về dạng đầy đủ.

b) Vẽ dạng bìa các nô của F.

c) Viết biểu thức rút gọn của F.

1.23. Hãy rút gọn các hàm 3 biến có bìa các nô dưới đây:

C \ AB	00 01 11 10			
	0	1	1	1
1	1	1		1

F_1

C \ AB	00 01 11 10			
	0	1	1	
1			1	1

F_2

$X_3 \backslash X_1 X_2$	00 01 11 10			
	0	1	1	1
1		1	1	

F_3

$X_3 \backslash X_1 X_2$	00 01 11 10			
	0	1	1	1
1		1		1

F_4

1.24. Hãy rút gọn các hàm sau bằng bìa các nô:

a) $F_1 = \sum m(2, 3, 8, 12)$

b) $F_2 = \sum m(0, 2, 3, 8, 10, 12)$

c) $F_3 = \sum m(0, 1, 3, 5, 6, 9, 11, 12, 13, 15)$

d) $F_4 = \sum m(0, 2, 3, 4, 5, 7, 8, 9, 13, 15)$

1.25. Cho các hàm logic 4 biến có bìa các nô đã biết dưới đây, hãy rút gọn các hàm này.

AB \ CD	00	01	11	10
00	1			1
01		1	1	
11		1	1	
10	1			1

F_1

AB \ CD	00	01	11	10
00	1	1	1	1
01	1			1
11	1			1
10	1	1	1	1

F_3

AB \ CD	00	01	11	10
00	1	1	1	1
01				1
11	1	1	1	1
10				1

F_5

AB \ CD	00	01	11	10
00		1	1	
01	1			1
11	1			1
10		1	1	

F_2

AB \ CD	00	01	11	10
00	1			
01	1	1	1	1
11	1			
10	1	1	1	1

F_4

AB \ CD	00	01	11	10
00	1	1		1
01		1		1
11		1		1
10	1	1		1

F_6

1.26. Hãy chứng minh các công thức sau :

a) $A.B + A.\bar{B} = A$

b) $A + AB = A$

c) $A + \bar{A}.B = A + B$

d) $AB + \bar{A}C + BC = AB + \bar{A}C$

e) $\overline{AB + \bar{A}.B} = \bar{A}.\bar{B} + AB$

f) $\overline{A.B + \bar{A}.C} = A.\bar{B} + \bar{A}.\bar{C}$

g) $A.(\bar{A} + B) = A.B$

h) $(A + B)(\bar{A} + C)(B + C) = (A + B)(\bar{A} + C)$

Có nhận xét gì về cặp công thức (c) với (g) và cặp (d) với (h).

Chương 2

CÁC CỔNG LOGIC CƠ BẢN VÀ MẠCH ĐIỆN CỔNG

Các mạch điện tử số sẽ hoạt động ở chế độ hai trạng thái (chế độ nhị phân), ở đó mỗi điện thế ở cổng vào và cổng ra của chúng chỉ nhận các giá trị 1 (có điện thế hoặc mức thế cao) và giá trị 0 (không có điện thế hoặc mức thế thấp). Đặc điểm quan trọng này cho phép chúng sẽ thực hiện được các hàm logic trong đại số logic khi thực hiện thiết kế mạch (tìm các mạch logic thích hợp để thực hiện một hàm logic muốn có) hoặc sử dụng đại số logic làm công cụ để mô tả, phân tích nhiệm vụ chức năng của chúng, qua đó phân tích, thiết kế các hệ thống kỹ thuật số.

Các cổng logic cơ bản là các mạch điện tử số quan trọng nhất để thực hiện các hàm logic sơ cấp đã cho trong bảng 1.4 chương 1. Sau khi đã nắm vững các cổng cơ bản, có thể thực hiện kết hợp chúng để tạo ra các hàm logic phức tạp hơn.

2.1. CÁC TÍNH CHẤT ĐẶC TRƯNG CỦA MỘT CỔNG LOGIC

Các nhà sản xuất bao giờ cũng cung cấp các tính chất đặc trưng của một cổng logic nhằm sử dụng các cổng này một cách có hiệu quả, đúng chức năng và khả năng của nó.

1. Loại cổng cơ bản chỉ ra hàm logic sơ cấp mà nó thực hiện tương ứng với một loại công nghệ chế tạo nó. Đó là các cổng thực hiện các hàm sơ cấp thể hiện trong bảng 1.4, ví dụ :

Hàm $F_4 = \overline{A + B}$ (hàm NOR) công nghệ RTL.

Hàm $F_3 = \overline{A.B}$ (hàm NAND) công nghệ DTL, TTL hay PMOS.

Hàm $F_2 = A + B$ và hàm $F_4 = \overline{A + B}$ công nghệ ECL.

Hàm $F_3 = \overline{A.B}$ và hàm $F_4 = \overline{A + B}$ công nghệ CMOS.

(Các loại công nghệ RTL thể hiện cấu tạo cổng gồm điện trở – tranzitor DTL gồm điốt – tranzitor, TTL gồm tranzitor – tranzitor... sẽ được đề cập tới ở các tiết sau).

2. Trở kháng tại đầu ra của cổng khi ở mức “1” và khi ở mức 0, người ta mong muốn thiết kế để cổng có trở kháng ra nhỏ.

3. Hệ số mắc tải: là số cổng logic khác có thể nối đồng thời tới đầu ra của cổng đang xét, chỉ xác định cho các cổng cùng một loại (thực hiện cùng một công nghệ và một hàm sơ cấp).

4. Nguồn một chiều dùng để cung cấp cho cổng làm việc.

5. Công suất tiêu thụ của một cổng logic được xác định theo số lượng các linh kiện (điện trở, tranzitor) làm việc ở chế độ bão hoà.

Bảng 2.1. CÁC ĐẶC TÍNH TIÊU CHUẨN CỦA CÁC CỔNG LOGIC

Nº	Loại công nghệ	RTL	DTL	HTL	TTL	ECL	PMOS	CMOS
	Tham số							
1	Loại cổng cơ bản	NOR	NAND	NAND	NAND	OR-NOR	NAND	NOR-NAND
2	Trở kháng ra (khi đầu ra ở mức cao), Ω	640	6K/2K	15K/1,5K	70	15	2K	1,5K
3	Trở kháng ra (khi đầu ra ở mức thấp), Ω	R_{bh}	R_{bh}	R_{bh}	R_{bh}	15	25K	1,5K
4	Nguồn cung cấp (V)	$3,6 \pm 10\%$	$5 \pm 10\%$	15 ± 1	$5 \pm 10\%$	$-52 \pm 10\%$	-27 ± 2 -13 ± 1	$+3 \pm 18$
5	Công suất tiêu thụ với một cổng (mW)	12	8	22	10	40	0,1	0,1-1
6	Mức độ chống nhiễu	thường	tốt	tốt	tốt	tốt	khá	rất tốt
7	Thời gian trễ với một cổng (ns)	12	30	90	10	2	700	30
8	Tần số xung nhịp cực đại (MHz)	8	12	4	35	60	1	10
9	Hệ số mắc tải	5	8	10	10	25	...	>50

6. Mức độ chống tạp âm (nhiều) xác định bởi biên độ điện áp nhiễu tối đa tác động tới đầu vào của cổng mà không làm thay đổi trạng thái đầu ra của nó. Được đánh giá theo hai tiêu chuẩn:

– Trạng thái hoạt động kém nhất (về điện thế nguồn, về hệ số tải, nhiệt độ...).

– Ảnh hưởng qua lại giữa các cổng logic cùng loại khi ghép nối với nhau (gây nhiễu cho nhau).

7. Thời gian trễ với một cổng là khoảng thời gian trôi qua từ lúc cổng nhận được tín hiệu logic ở đầu vào cho đến khi đầu ra của nó xác lập được trạng thái logic tương ứng. Đối với mỗi cổng logic, thời gian này thường nhỏ (vài chục phần tỷ giây) nhưng cần chú ý khi phải ghép nối nhiều cổng lại. Thường người ta xác định thời gian trễ sau khi tín hiệu logic truyền qua 1 cặp hai cổng tức là lấy trung bình chia đều sự chậm trễ cho hai trạng thái lên cao và xuống thấp.

8. Tần số xung nhịp cực đại là khả năng phản ứng nhanh nhất của một cổng mà nó vẫn còn phân biệt tốt hai trạng thái logic. Xác định tần số này nhờ mắc 1 cặp cổng tạo thành mạch tạo xung nhịp có tần số xung tạo ra cao tới đâu.

Các đặc tính đã nêu có thể tóm tắt lại trong bảng 2.1 ở đó có đưa ra các giá trị cụ thể tiêu chuẩn của các tham số này.

2.2. CỔNG THỰC HIỆN HÀM CỘNG LOGIC (cổng OR)

$$F_{OR} = A + B \quad (2.1)$$

2.2.1. Bảng trạng thái, biểu thức hàm, các ký hiệu quy ước của cổng OR (hình 2.1a,b), đồ thị xung điện áp theo thời gian (hình 2.1c) của cổng OR 2 lối vào

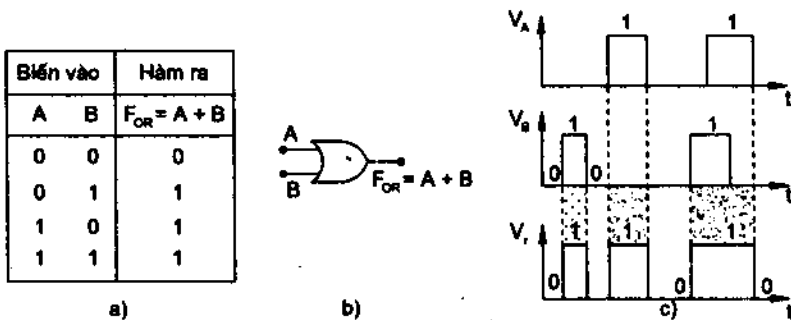
Ta nhận xét là:

$F_{OR} = 1$ khi có bất kỳ đầu vào nào của nó có trị 1

$F_{OR} = 0$ chỉ khi tất cả các đầu vào có trị 0

$F_{OR} = 1$ khi tất cả các đầu vào có trị 1 tức là có tính chất

$$A + B + \dots + N = 1 + 1 + \dots + 1 = 1$$



Hình 2.1. a) Bảng trạng thái hàm OR ; b) ký hiệu quy ước ; c) Đồ thị thời gian minh họa

2.2.2. Ví dụ

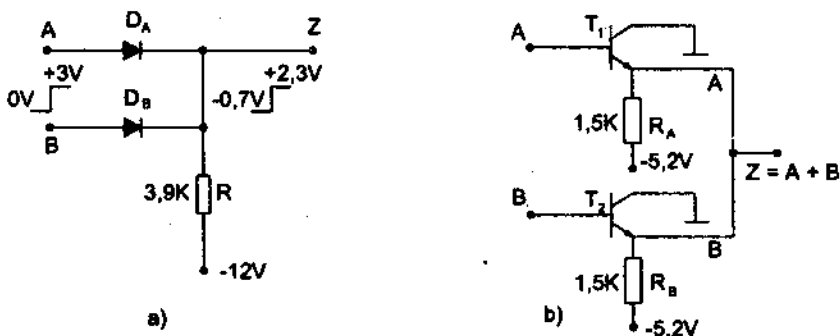
Một vài dạng mạch điện tử số thực hiện hàm F_{OR} được cho trên hình 2.2a,b.

* Ở mạch 2.2a khi $A = 1$ hoặc $B = 1$ hoặc $A = B = 1$ (tương ứng với mức điện thế +3V) ta nhận được $Z = 1$ (ứng với mức điện thế +2,3V).

Còn khi $A = B = 0$ (mức điện thế gần 0V) cổng ra ở mức -0,7V ($Z = 0$).

* Ở mạch 2.2b, T_1 và T_2 là 2 tranzitor tại 2 cổng ra của 2 vi mạch số chế tạo theo công nghệ logic ghép emitter (logic không bão hòa - ECL) khi đầu hai đầu ra của chúng với nhau ta được $Z = A + B$. Khi sử dụng loại ECL có thể chỉ dùng một IC với 2 cổng vào A và B và 2 cổng ra trong đó có một cổng ra thực hiện hàm Z (cổng ra còn lại thực hiện hàm $\bar{Z}$).

* Công nghệ chế tạo cổng OR thường khó khăn hơn so với các loại cổng khác vì những lý do kỹ thuật thực hiện.



Hình 2.2. Ví dụ về mạch điện tử số thực hiện cổng OR

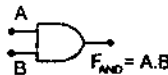
2.3. CÔNG THỨC HIỆN HÀM NHÂN LOGIC (công AND)

$$F_{AND} = A.B \quad (2.2)$$

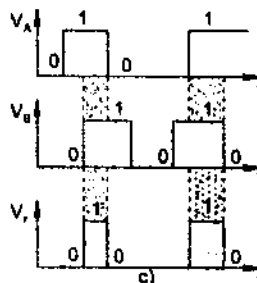
2.3.1. Bảng trạng thái, biểu thức hàm, ký hiệu quy ước và giản đồ xung điện áp theo thời gian của cổng AND 2 lối vào (cho trên hình 2.3)

Biến vào		Biến ra
A	B	$F_{AND} = A.B$
0	0	0
0	1	0
1	0	0
1	1	1

a)



b)



c)

Hình 2.3. a) Bảng chân lý của cổng AND ; b) Ký hiệu quy ước ; c) Đồ thị thời gian,

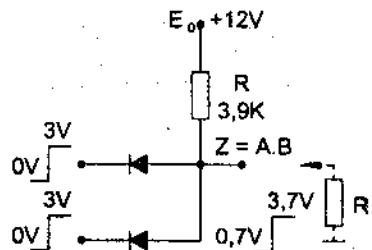
Ta có nhận xét là:

$F_{AND} = 1$ chỉ khi tất cả các cổng vào có giá trị logic 1.

$F_{AND} = 0$ khi có ít nhất một cổng vào có giá trị 0.

2.3.2. Mạch điện tử số thực hiện hàm F_{AND}

Hình 2.4 đưa ra mạch điện tử số thực hiện hàm F_{AND} . Khi có một đầu vào nào đó ở mức điện thế thấp, điốt tương ứng với đầu vào này sẽ dẫn điện, khi đó điện thế ở cổng ra (khi không tải) sẽ ở mức thấp bằng giá trị điện áp thuận rơi trên điốt (0,7V với loại điốt Si). Còn khi tất cả các cổng vào đều ở mức điện thế cao (ví dụ +3V) các điốt đều không dẫn điện làm giảm áp trên điện trở $R=3,9k\Omega$ nhỏ, Z đầu ra ở mức



Hình 2.4. Cổng AND dùng điốt bán dẫn

điện thế cao. Ta chú ý rằng khi mắc $R_{tải}$ ở tại cửa ra, R và $R_{tải}$ hình thành 1 bộ chia áp điện trở khi $A = B = 1$, khi đó cần đảm bảo điều kiện của mức ra cao nhỏ nhất:

$$U_{\text{ra min}} = \frac{E_o}{R + R_{\text{tải}}} R_{\text{tải}} = \frac{12V}{3,9K + R_{\text{tải}}} R_{\text{tải}} = 2V$$

$$\text{hay } 12V \cdot R_{\text{tải}} = 2V(3,9K + R_{\text{tải}})$$

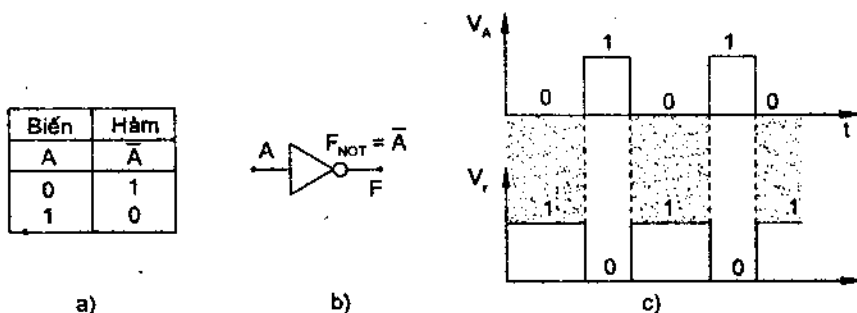
suy ra điều kiện đối với tải mắc vào cổng là $R_{\text{tải min}} = 780\Omega$

Cũng như cổng OR, việc thực hiện bằng các mạch điện tử số cổng AND không được thuận lợi vì lý do công nghệ.

2.4. CỔNG THỰC HIỆN HÀM ĐẢO (phủ định logic – NOT)

$$F_{\text{NOT}} = \bar{A} \quad (2.3)$$

2.4.1. Bảng chân lý, ký hiệu quy ước và giản đồ thời gian của cổng NOT (được cho trên hình 2.5)

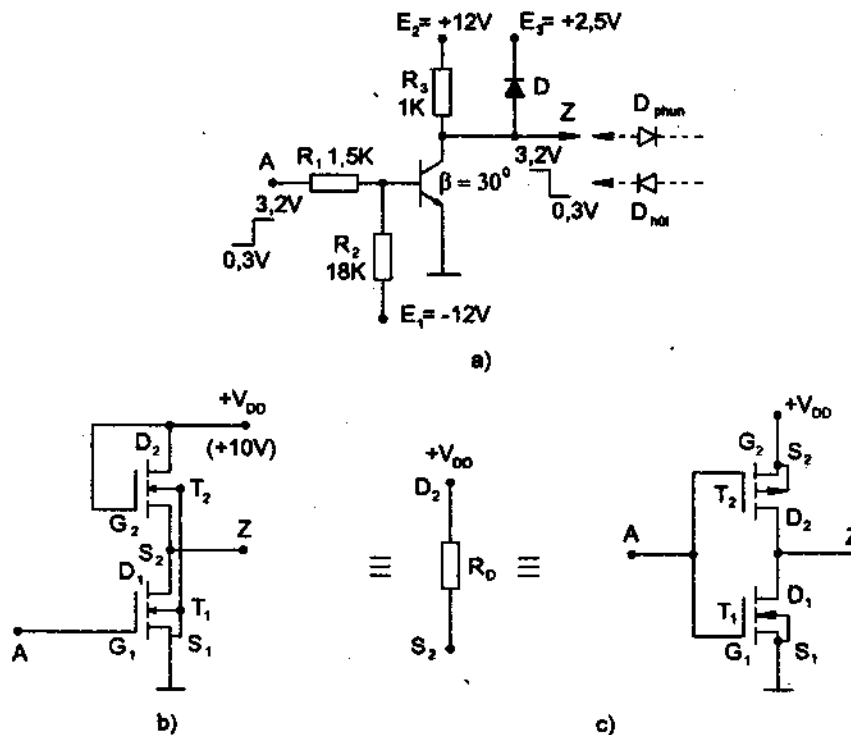


Hình 2.5. Bảng chân lý (a); Ký hiệu quy ước (b) và đồ thị thời gian (c) của cổng NOT.

Như vậy cổng NOT luôn luôn chỉ có một đầu vào và một đầu ra với giá trị biến vào và hàm ra luôn là giá trị đảo của nhau. Khi ghép liên tiếp 2 cổng NOT ta sẽ nhận được hàm lặp (hàm G_4 trong hình 1.5 ở chương 1), trị hàm ra luôn bằng trị biến vào.

2.4.2. Mạch số thực hiện cổng NOT (hình 2.6)

1. Ở hình 2.6a, T làm việc ở chế độ đóng ngắt, khi V_A ở mức thấp thì T ngắt mạch (không dẫn dòng điện) điện thế cổng ra V_Z ở mức cao. Khi V_A chuyển lên mức cao ($A = 1$) thì T nối mạch ở chế độ bão hoà, giảm áp trên R_3 lớn chuyển V_Z về mức thấp ($Z = 0$).



Hình 2.6. Các dạng cổng NOT : a) Dùng RTL ; b) Dùng MOS kênh N cảm ứng ; c) Dùng CMOS kênh cảm ứng...

* Nguồn $E_1 < 0$ đảm bảo cho T ngắt tin cậy khi V_A ở mức thấp (T hở mạch chắc chắn). Nguồn $E_2 > 0$ và $E_3 > 0$ có tác dụng giữ cho mức cao ở đầu ra ở giá trị quy định bằng $E_3 + U_D = 3,2V$. Khi $A = 1$ tương ứng với mức vào $V_A = +3,2V$ thì T dẫn điện.

* Cần lựa chọn các tham số của mạch sao cho T lúc dẫn điện đạt tới chế độ bão hoà:

Ví dụ trong mạch 2.6a đang xét, dòng colectơ bão hoà bằng:

$$I_{CS} = \frac{E_2}{R_3} = \frac{12V}{1k\Omega} = 12mA$$

từ đó suy ra dòng bazơ bão hoà:

$$I_{BS} = \frac{I_{CS}}{\beta} = \frac{12mA}{30} = 0,4mA$$

Với các tham số đã cho, khi $V_A = +3,2V$ thì đạt được $I_B = 1mA$ nghĩa là $I_B > I_{BS}$ đảm bảo điều kiện cho T ở bão hoà với khái niệm độ sâu bão hoà của T được xác định bởi:

$$\frac{I_B}{I_{BS}} = \frac{1mA}{0,4mA} = 2,5$$

* Khi tải của mạch hình 2.6a được nối tới cổng ra qua một diốt chọn $\Phi_{hút}$ hay Φ_{phun} (đường vẽ đứt nét) ta nói cổng NOT khi đó làm việc tương ứng ở chế độ hút dòng điện hay chế độ phun dòng điện.

2. Trong mạch hình 2.6b cổng NOT được thực hiện nhờ công nghệ NMOS loại kênh cảm ứng (kênh chỉ có khi có tín hiệu thích hợp cực tính dương xuất hiện ở đầu vào, các vi điện tử số dựa trên cấu trúc này ký hiệu là E/EMOS).

* Cổng gồm 2 MOSFET kênh N cảm ứng T_1 là phần tử khuếch đại ở chế độ chuyển mạch (đóng, ngắt), T_2 làm nhiệm vụ phụ tải thay điện trở R_D (có điện thế U_{DG} luôn bằng 0 nên luôn làm việc ở chế độ bão hoà). Giả thiết điện áp mở của T_1 là $V_p = +2V$, khi $A = 0$ (tức là $V_A = 0$) T_1 ngắt mạch khi đó $V_Z = V_{DD} - V_p = 10V - 2V = 8V$ tương ứng với giá trị $Z = 1$. Khi $V_A = 8V$ ($A = 1$) T_1 nối mạch với điện trở nối mạch $R_{DS1} \ll R_{DS2}$ thì $V_Z = 0V$ hay $Z = 0$.

* Khi nối cực G_2 của T_2 tới 1 nguồn $V_0 > V_{DD}$, ví dụ $V_0 = 14V$ thì T_2 ở chế độ không bão hoà và mạch tương ứng có tên loại cổng NOT không bão hoà với tốc độ chuyển mạch được tăng lên rõ rệt.

3. Mạch hình 2.6c là một dạng cổng NOT dùng công nghệ CMOS (ở đó cặp MOSFET kênh cảm ứng T_1 loại kênh N, T_2 loại kênh P được chế tạo kiểu sinh đôi). Trên hình 2.6b và c nếu không sử dụng T_2 làm điện trở tải, người ta dùng điện trở R_D mắc ở cực máng của T_1 , R_D có giá trị vài chục $k\Omega$ và giá trị R_D lớn đảm bảo đặc tính chuyển mạch lúc tính tốt do công suất tiêu thụ lúc T_1 nối mạch nhỏ và mức điện thế thấp ($Z = 0$) ứng với V_Z có giá trị nhỏ, việc chuyển trạng thái của T_1 có độ dốc từ mức 1 về mức 0 lớn nên thời gian trễ nhỏ, chống nhiễu tốt. Tuy nhiên khi tải có tính điện dung ($C_{tải}$) (thường gặp trong thực tế đặc biệt với nhóm công nghệ

MOS, khi ghép liên tiếp các cổng MOS lại), do quá trình nạp của $C_{tải}$ qua R_D từ mức $Z = 0$ đến mức $Z = 1$ xảy ra chậm vì R_D lớn, gây ảnh hưởng nghiêm trọng tới việc chuyển mức từ thấp tới cao của V_Z (chuyển giá trị của Z từ 0 lên 1). Vì các lý do trên, người ta thay thế R_D bằng tải là T_2 như hình 2.6b và c.

* Trong mạch 2.6c, T_2 là loại MOS kênh P cảm ứng, các cực của G_1 và G_2 nối chung và làm đầu vào, các cực D_1 và D_2 nối chung thành đầu ra, khi làm việc S_2 nối cực dương nguồn $+V_{DD}$ còn S_1 nối với 0V. Chọn $V_{DD} > V_{P1} + |V_{P2}|$ (V_{P1} và V_{P2} là các ngưỡng điện áp mở của T_1 và T_2). Khi $A = 0$ (V_A ở mức thấp) T_1 ngắt mạch do đó $|V_{GS2}| = |-V_{DD}| > |V_{P2}|$ làm T_2 nối mạch, điện thế đầu ra ở mức cao $Z = 1$. Còn khi $A = 1$ (V_A ở mức cao hơn giá trị V_{P1}) T_1 nối mạch làm $V_{GS2} = 0$ và T_2 ngắt mạch điện thế đầu ra ở mức thấp ứng với $Z = 0$.

* Như vậy với mọi giá trị A (bằng 0 hay bằng 1) trong cặp T_1 T_2 luôn có một phân tử ngắt mạch, dòng điện tĩnh đi qua chúng xấp xỉ 0 và công suất tiêu hao ở chế độ tĩnh đạt nhỏ tới cỡ 10^{-6} đến 10^{-7} W rất có ưu điểm trong việc chế tạo tích hợp chúng với mật độ lớn thành IC số, đây là công nghệ tiên tiến nhất của IC số.

2.5. CỔNG THỰC HIỆN HÀM LOGIC CỘNG ĐẢO

(hay hoặc đảo – cổng NOR)

$$F_{NOR} = \overline{A+B} \quad (2.4)$$

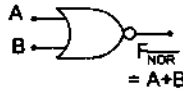
2.5.1. Bảng chân lý, ký hiệu quy ước và giản đồ điện áp theo thời gian của một cổng NOR có 2 đầu vào (hình 2.7 a,b,c)

Ta có nhận xét là:

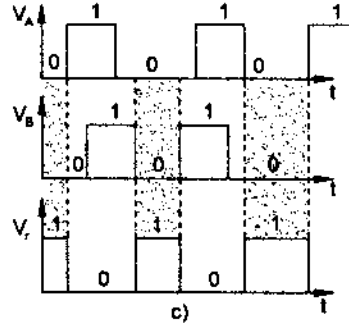
- Đầu ra cổng NOR sẽ lên mức cao ($F_{NOR} = 1$) khi tất cả các đầu vào của nó ở mức thấp.
- $F_{NOR} = 0$ (đầu ra ở mức thấp) khi có ít nhất 1 đầu vào của nó ở mức cao.
- Cổng NOR là sự kết hợp liên tiếp cổng OR và cổng NOT (hình 2.7d).

Biến vào		Hàm ra
A	B	$F_{NOR} = A + B$
0	0	1
0	1	0
1	0	0
1	1	0

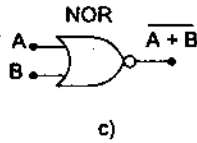
a)



b)

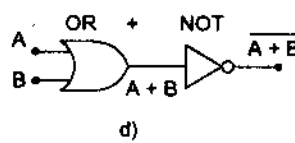


c)



c)

≡



d)

Hình 2.7. Cổng NOR

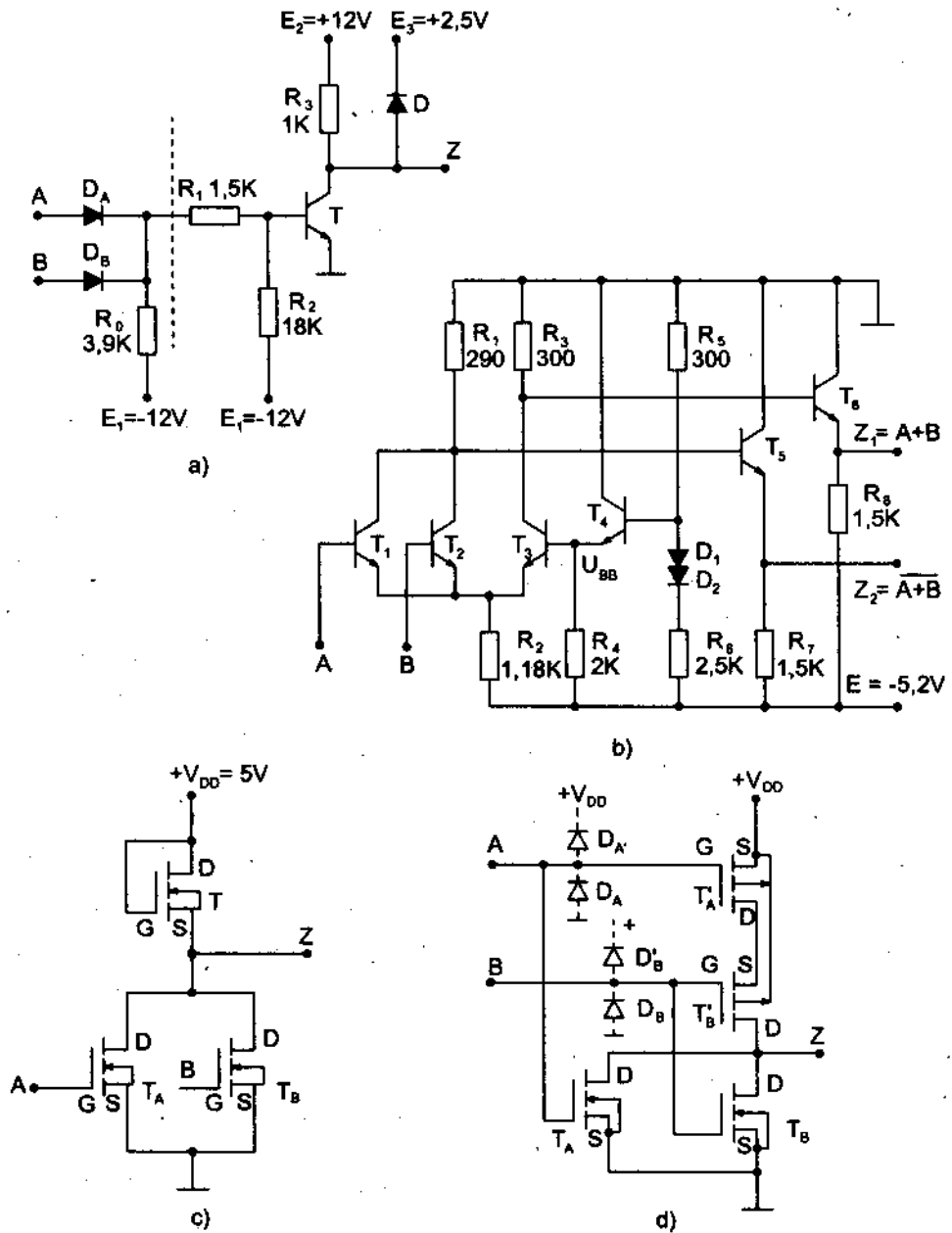
a) Bảng chân lý ; b) Ký hiệu quy ước ; c) Giải đồ thời gian minh họa.

2.5.2. Cổng logic thực hiện hàm NOR – cổng NOR (hình 2.8)

– Mạch hình 2.8a là sự ghép nối tiếp cổng OR hình 2.2a và cổng NOT hình 2.6a theo ký hiệu tương đương hình 2.7d vừa nêu trên.

– Mạch hình 2.8b là cổng NOR – OR theo công nghệ ECL (logic không bão hoà) có tốc độ làm việc thuộc nhóm nhanh nhất trong các loại cổng logic. T_1 và T_2 là tranzitor đầu vào, T_3 là tranzitor chuẩn kết hợp với T_1 hoặc T_2 tạo tính chất vi sai, T_4 kết hợp bộ chia áp R_5 , R_6 và 2 diốt để ổn định nhiệt D_1 , D_2 có nhiệm vụ tạo ra 1 điện áp chuẩn có giá trị $-1,175V$ tại cực emitor của T_4 $U_{BB} = -1,175V$ (đây là giá trị điện áp đã được chọn kỹ để thỏa mãn mức logic và chống tạp âm), T_5 và T_6 là các tranzitor ở đầu ra Z_2 và Z_1 tương ứng.

Khi A và B ở điện thế rất thấp ($< -1,25V$ tương ứng $A = B = 0$) T_1 và T_2 ngừng dẫn điện, T_3 dẫn dòng (nhưng không bão hoà) có $V_{c3} \approx -1V$ và $V_{E3} \approx 1,9V$ (do U_{BB} duy trì) đầu ra Z_1 khi đó ở mức thấp: $V_{c3} - V_{BE6} \approx -1,65V$ đồng thời đầu ra $Z_2 = A + B$ ở mức điện thế cao ($-0,75V$) vì $V_{c1} = V_{c2} = 0V$.



Hình 2.8. Các dạng cổng NOR

- a) Cổng NOR kết hợp từ hai cổng OR và NOT
- b) Cổng OR- NOR trong công nghệ logic ghép emitter (ECL - không bão hoà)
- c) Cổng NOR chế tạo theo công nghệ N-MOS
- d) Cổng NOR chế tạo theo công nghệ CMOS

Khi có 1 đầu vào (ví dụ A) có điện thế bất âm đi ($A=1$) T_1 dẫn điện vì đủ điện thế V_{BE1} dòng $I_{c1} \uparrow I_{c3} \downarrow$ dẫn tới $V_{c1} \downarrow V_{c3} \uparrow$ tới lúc T_3 ngừng dẫn (khi V_A đạt cỡ $-1V$) đầu ra Z_1 nâng lên mức $-0,75V$ ($Z_1 = A + B = 1$), còn đầu ra $Z_2 = \overline{A+B}$ có điện thế giảm tới $-1,65V$ ứng với giá trị $Z_2 = 0$. Hoàn toàn tương tự với $B = 1$ hay $A = B = 1$, đó là bảng chân lý của cổng NOR (với Z_2) và cổng OR (với Z_1).

Nhờ cặp T_5, T_6 , tổng điện trở ra của mạch (mắc CC) nhỏ cỡ 15Ω nhờ đó tốc độ tác động nhanh, có hai đầu ra liên hợp thuận tiện cho việc sử dụng, có tổng trở đầu vào lớn (cỡ $100k\Omega$). Tuy nhiên cấu trúc 2.8b dùng nguồn âm, mạch phức tạp và khó phối kết hợp với các cổng logic khác.

– Mạch hình 2.8d sử dụng 2 cặp FET T_A, T'_A và T_B, T'_B 2 tranzitor trong một cặp làm việc ở chế độ luân phiên nhau, được chế tạo theo công nghệ CMOS. Mạch làm việc như sau: khi có ít nhất một đầu vào ở mức cao (ví dụ $A=1$) T_A nối mạch và Z ở mức thấp ($Z = 0$). Khi $A = B = 0$ cả T_A và T_B ngắt mạch Z chuyển lên mức thế cao ($Z = 1$). Cổng hình 2.8c hoạt động tuân theo quy luật của bảng chân lý 2.7a đối với hàm NOR.

Đường nét đứt ở các đầu vào A, B sử dụng các diốt D_A, D'_A và D_B, D'_B có nhiệm vụ bảo vệ cửa vào chống bị đánh thủng lớp điện môi cách ly cực G và kênh hay đánh thủng tĩnh điện với lớp này.

2.6. CỔNG LOGIC THỰC HIỆN HÀM LOGIC NHÂN- ĐẢO (cổng NAND)

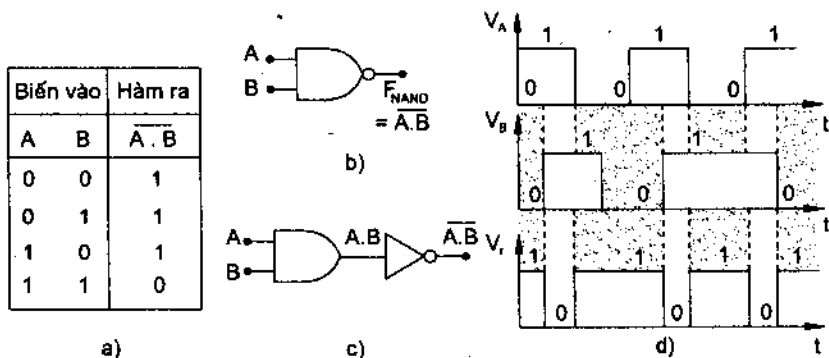
$$F_{NAND} = \overline{A.B} \quad (2.5)$$

2.6.1. Bảng chân lý, ký hiệu quy ước và giản đồ thời gian của một cổng NAND hai đầu vào (hình 2.9a,b và d), ký hiệu tương đương của cổng NAND (hình 2.9c)

Ta có các nhận xét sau :

- $F_{NAND} = 0$ chỉ khi tất cả các đầu vào của cổng NAND ở mức cao ($A = B = 1$).
- $F_{NAND} = 1$ khi có ít nhất một đầu vào của cổng NAND ở mức thấp.

– Có thể xem cổng NAND là việc ghép nối tiếp một cổng AND với một cổng NOT (hình 2.9c).

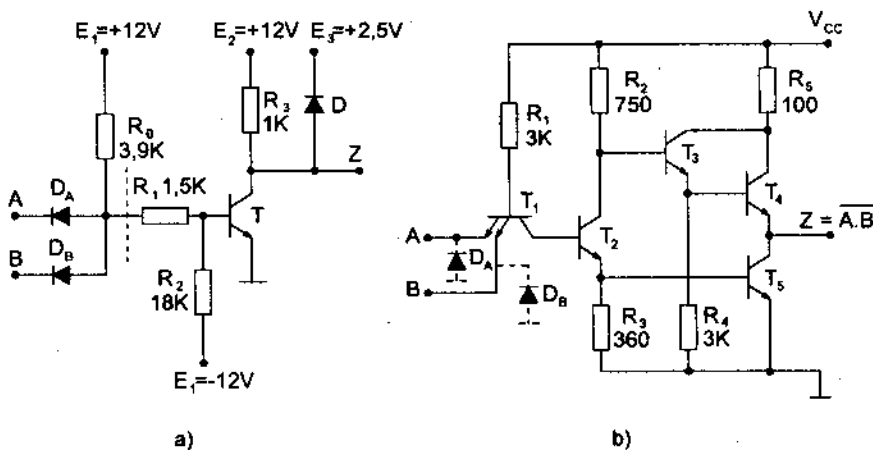


Hình 2.9. Cổng NAND

a) Bảng chân lý ; b), c) Ký hiệu quy ước và ký hiệu tương đương ;
d) Đồ thị thời gian minh họa bảng chân lý

2.6.2. Cấu trúc các mạch điện tử số thực hiện các cổng NAND từ các công nghệ số DTL và TTL (hình 2.10)

– Mạch điện hình 2.10a mô tả cấu trúc cổng NAND DTL nhờ cách ghép nối tiếp cổng AND hình 2.4 với một cổng NOT hình 2.6a thực hiện theo cấu trúc mô tả bởi hình 2.9c.

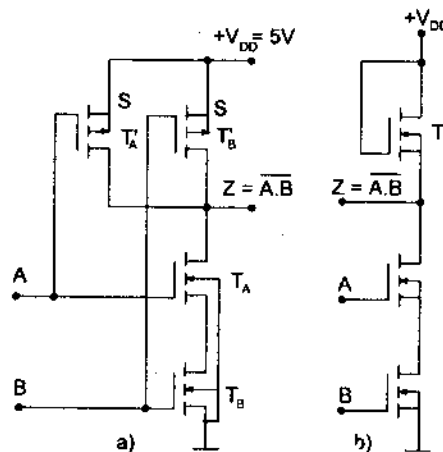


Hình 2.10 a) Cổng NAND kết hợp liên tiếp cổng AND và NOT (công nghệ DTL)
b) Cổng NAND loại TTL.

– Mạch điện hình 2.10b là cấu trúc cổng NAND loại TTL gồm 3 phần chính: T_1 (loại tranzitor nhiều cực emitor) kết hợp R_1 có chức năng một cổng AND có 2 cửa vào A và B, T_2 kết hợp R_2 và R_3 tạo ra một tầng khuếch đại đệm đảo pha phân tải chế độ đóng ngắt để làm nguồn tín hiệu điều khiển mạch điện tăng ra T_4 và T_5 làm việc ở chế độ đóng ngắt kiểu đẩy kéo (luân phiên nhau) nhờ hai tín hiệu ngược pha nhau xuất hiện trên colectơ và emitor của T_2 . Khi có ít nhất một đầu vào ở mức thấp, T_1 sẽ dẫn điện bão hoà theo đầu vào này kéo theo T_2 và T_5 ngắt mạch, T_3 và T_4 nối mạch đầu ra ở mức cao ($Z = 1$). Khi tất cả đầu vào ở mức cao ($A = B = 1$) T_1 ngắt mạch đẩy T_2 và T_5 nối mạch, T_3 và T_4 ngắt mạch, đầu ra ở mức thấp ($Z = 0$) chính là chức năng của cổng NAND theo bảng chân lý 2.9a. Việc tính toán chi tiết với giả thiết khi A ở mức thấp $V_A = 0,3V$ thì $V_{B1} = 1V$ và $I_{B1} = 1,33mA$, T_1 bão hoà sâu với $V_{CE1} \approx 0,1V$ và $V_{B2} = V_A + V_{CE1} = 0,4V$ làm T_2 và T_5 ngắt, T_3 và T_4 nối mạch I_{B3} rất nhỏ nên $V_{B3} \approx 5V$, do đó $V_Z = V_{B3} - V_{BE4} - V_{BE3} = 5V - 1,4V = 3,6V$ tức là Z ở mức cao ($Z = 1$).

– Để nâng cao chất lượng cổng NAND (với vai trò là loại cổng phổ biến nhất trong các cổng cơ bản đã nêu) về các mặt: tốc độ chuyển trạng thái nhanh, tiêu hao năng lượng giảm, năng lực chống nhiễu tốt và tăng mật độ thích hợp phối ghép trong IC số, người ta tìm cách cải tiến mạch hình 2.10b theo hướng: chống bão hoà cho T_1 , T_2 , T_3 và T_5 nhờ diốt Shotky phân dòng I_B cho các tranzitor, bổ sung một tranzitor T_5 làm nguồn phóng điện cho bazơ của T_5 nhờ đó tăng độ dốc của đặc tính truyền đạt điện áp (xem thêm tài liệu tham khảo).

2.6.3. Mạch số hình 2.11 là cấu trúc các cổng NAND họ FET theo công nghệ CMOS và NMOS. Ở mạch 2.11a T_A và T_B là loại MOS kênh N điều khiển bằng điện thế dương, T'_A và T'_B loại kênh p, cặp $T_A T'_A$ và $T_B T'_B$ chế tạo song sinh từng cặp làm việc đóng ngắt kiểu đẩy kéo. Khi $A = B = 1$ (ở mức cao) T_A và T_B đều nối mạch làm thế đầu ra xuống mức thấp ($Z = 0$), khi có ít nhất một trong các đầu vào ở



Hình 2.11

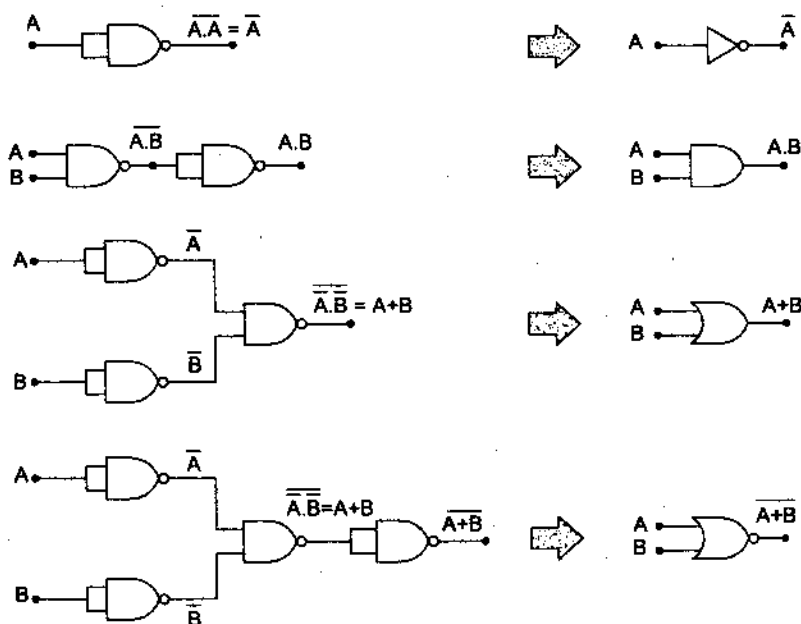
Cấu trúc cổng NAND

a) Công nghệ CMOS ; b) Công nghệ NMOS.

thế thấp (ví dụ $A = 0$). T_A ngắt mạch và trên mạch nối tiếp T_A, T_B không có dòng điện, lõi ra dâng lên mức cao ($Z = 1$), đó chính là chức năng cổng NAND theo bảng chân lý 2.9a. Mạch 2.11b hoạt động tương tự trên, T luôn dẫn điện bão hoà (vì cực G nối với cực D) có vai trò là tải của T_A và T_B .

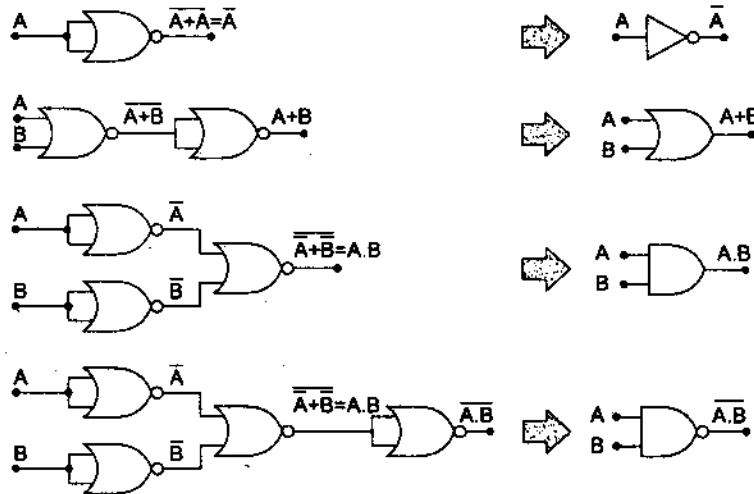
2.7. TÍNH CHẤT ĐA DỤNG CỦA CỔNG NAND VÀ CỔNG NOR

2.7.1. Theo tính chất của một hệ hàm cơ sở đầy đủ đã xét ở chương 1, mọi hàm logic đều có thể thực hiện được nhờ cách kết hợp 3 hàm cơ sở OR, AND và NOT, tức là mọi cấu trúc mạch logic phức tạp đều có thể tổng hợp được từ hệ hàm đầy đủ này (chỉ xét các mạch logic tổ hợp, các hệ logic dãy sẽ được xét ở chương sau). Tuy nhiên do đặc điểm công nghệ chế tạo các cổng OR, AND và NOT có nhiều điểm khác nhau và nói chung khó thực hiện trong các mạch tích hợp vi điện tử số do tính không đồng nhất này.



Hình 2.12. Tính đa dụng của cổng NAND

2.7.2. Các cổng logic NAND và NOR là loại cổng đa dụng : chỉ từ 1 loại cổng NAND hoặc từ một loại cổng NOR, ta có thể xây dựng được các cổng cơ bản còn lại và do đó các cổng này tự mỗi cổng đã tạo được một hệ đầy đủ. Hình 2.12 mô tả cách xây dựng các cổng cơ bản NOT, OR, AND và NOR từ chỉ một loại cổng NAND và hình 2.13 mô tả cách xây dựng các cổng NOT, OR, AND và NAND từ một loại cổng NOR, từ đó thấy rõ tính đa dụng (thay thế được) của hai loại cổng trên.



Hình 2.13. Tính đa dụng của cổng NOR

Từ hình 2.12 và 2.13 ta có thể kết luận : mọi mạch logic tổ hợp đều có thể xây dựng chỉ từ một loại cổng cơ bản là cổng NAND hay chỉ từ một loại cổng NOR, đây là một tính chất đặc biệt quan trọng trên quan điểm tính chất đồng nhất trong công nghệ chế tạo từ đó giá thành chi phí giảm, độ tin cậy cao và tính lắp lẫn tối ưu.

2.7.3. Ví dụ minh họa tính chất đa dụng

Ví dụ 1: Cho hàm logic $Z = AB + \overline{A}\overline{B}$ (2.6)

Hãy xây dựng cấu trúc thực hiện hàm Z bằng các cổng NAND.

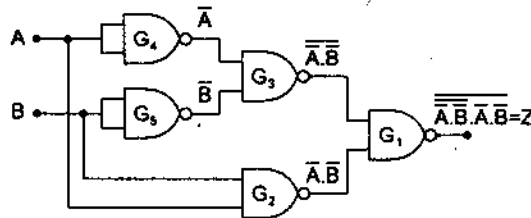
Áp dụng quy tắc (1.9) cho hàm Z, ta có:

$$Z = \overline{\overline{Z}} = \overline{\overline{AB + \overline{A}\overline{B}}}$$

Sử dụng luật De Morgan (1.14) cho biểu thức trên với dấu đảo ở dưới:

$$\begin{aligned} Z &= \overline{\overline{A.B.A.B}} \\ &= \overline{G_2.G_3} = G_1 \end{aligned}$$

từ đây, ta thấy rõ để có Z cần sử dụng 5 cổng NAND có cấu trúc như hình (2.14) với G_4 và G_5 thực hiện phép $\overline{A}$ và $\overline{B}$.

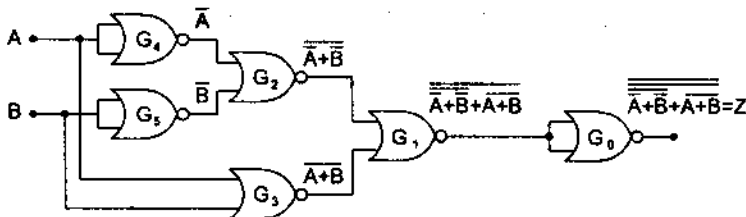


Hình 2.14. Cấu trúc thực hiện hàm $Z = AB + \overline{A}\overline{B}$ từ các cổng NAND 2 đầu vào

Ta có thể xây dựng cấu trúc thực hiện hàm Z đã cho bằng các cổng NOR, muốn vậy tiến hành như sau, với việc áp dụng (1.9) và (1.14):

$$\begin{aligned} A.B &= G_2 = \overline{\overline{A.B}} = \overline{A+B} \\ \overline{A.B} &= G_3 = \overline{\overline{\overline{A.B}}} = \overline{\overline{A+B}} = A+B \\ Z &= G_2 + G_3 = \overline{\overline{A+B}} + \overline{A+B} = \overline{G_1} \end{aligned}$$

Với cách đặt này, Z sẽ được biểu diễn dưới dạng chỉ gồm các cổng NOR như hình 2.15.



Hình 2.15. Cấu trúc thực hiện Z từ các cổng NOR 2 đầu vào

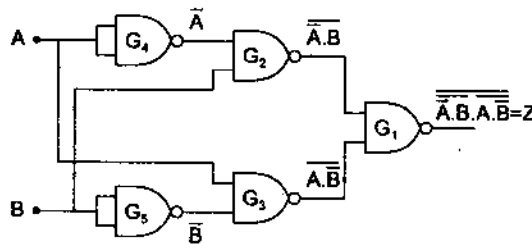
Ví dụ 2: Cho hàm $Z = \overline{A}.B + A.\overline{B}$ (xem hàm F7 của bảng 1.4) (2.7)

Hãy xây dựng cấu trúc mạch logic thực hiện Z chỉ từ các cổng NAND và chỉ từ các cổng NOR.

Ta viết lại hàm Z dưới dạng tương tự như cách làm ở ví dụ 1, có kết quả sau:

$$Z = \overline{\overline{\overline{A}.B + A.\overline{B}}} = \overline{\overline{A}.B} . \overline{A.\overline{B}}$$

từ kết quả này ta nhận được cấu trúc hình 2.16.

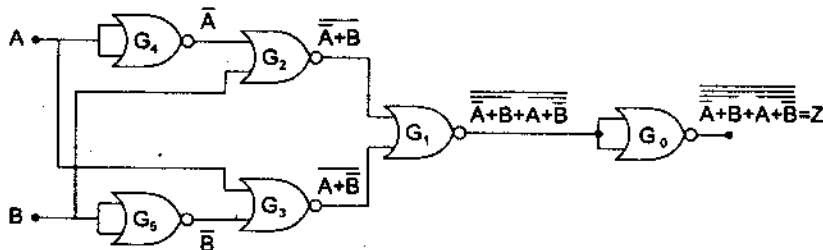


Hình 2.16. Cấu trúc hàm (2.7) từ các cổng NAND 2 đầu vào

Mặt khác nếu biểu diễn Z dưới dạng khác:

$$Z = \overline{\overline{\overline{A}.B} + \overline{A.\overline{B}}} = \overline{\overline{A}.B} + \overline{A.\overline{B}}$$

ta nhận được kết quả cấu trúc hình 2.17.



Hình 2.17. Cấu trúc hàm (2.7) từ các cổng NOR 2 đầu vào

Ví dụ 3: Băng tải của một quy trình sản xuất sẽ ngừng chạy khi gặp các sự cố sau (được giám sát bởi các cơ cấu tự động cảnh báo):

- Khi tốc độ chạy nhanh hơn một mức cho phép.
- Khi thùng chứa sản phẩm ở cuối băng bị đầy.
- Do ma sát tăng làm băng bị kéo quá căng.
- Khi người điều khiển tắt chế độ vận hành băng tay.

Hãy thiết lập hàm logic biểu thị tín hiệu thông báo hay điều khiển cảnh báo các sự cố trên.

Ta có nhận xét, nếu gán cho các trạng thái sự cố là A,B,C,D tương ứng với giá trị 0 khi không có sự cố, giá trị 1 khi xảy ra sự cố thì $Z_1 = A.B$ (hàm AND) thể hiện hai sự cố đầu tiên. $Z_2 = C.D$ thể hiện hai sự cố hoặc hành động tiếp sau. Khi hoặc $Z_1 = 1$ hoặc $Z_2 = 1$ thì băng ngừng hoạt động, vậy:

$$Z = Z_1 + Z_2 = A.B + C.D.$$

Cấu trúc Z có thể thực hiện nhờ 2 cổng AND và 1 cổng OR.

Nếu chỉ dùng các cổng NAND ta có:

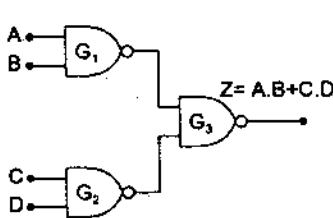
$$Z = \overline{\overline{A.B + C.D}} = \overline{\overline{A.B} . \overline{C.D}}$$

ta nhận được cấu trúc hình 2.18.

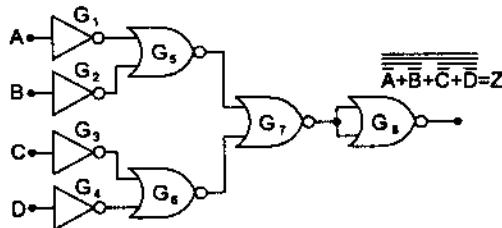
Còn khi chỉ dùng các cổng NOR ta nhận được cấu trúc hình 2.19.

$$Z = \overline{\overline{A.B + C.D}} = \overline{\overline{A + B + C + D}}$$

Qua ví dụ 3 ta thấy rõ việc sử dụng cổng NAND mang tính chất phổ biến hơn so với cổng NOR do kết cấu đơn giản và kinh tế hơn.



Hình 2.18



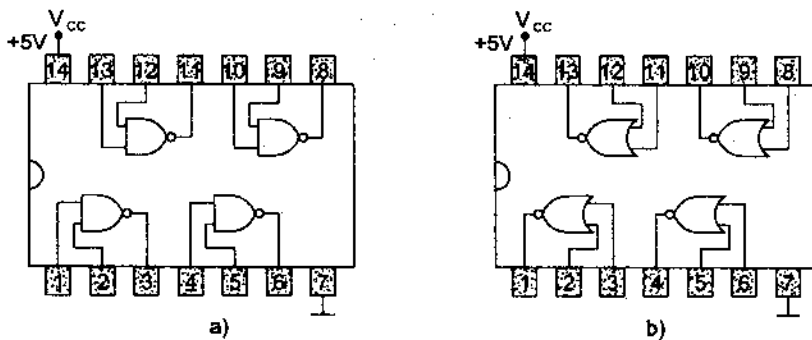
Hình 2.19

2.8. MỘT SỐ IC CỔNG THƯỜNG GẶP

Thực tế phổ biến nhất là nhóm IC cổng họ TTL và IC cổng họ CMOS, ta sẽ giới thiệu ở tiết này một số loại IC cổng thông dụng về cấu trúc khối, ký hiệu quy ước và bố trí các chân của chúng.

2.8.1. Các IC số loại TTL/ LS thực hiện các hàm logic cơ bản

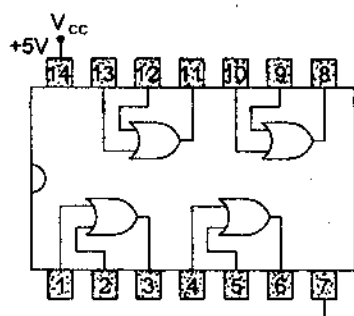
* Mỗi IC có 14 chân với 4 cổng NAND hoặc NOR riêng biệt bên trong như thể hiện trên các hình 2.20.



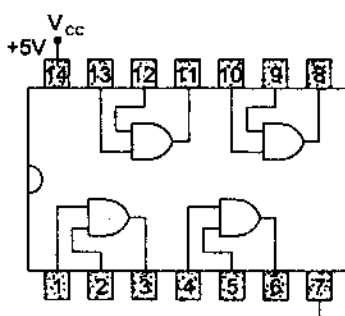
Hình 2.20. Sơ đồ bố trí chân IC TTL
a) Cổng NAND trong IC 7400 hay 74LS00 ;
b) Cổng NOR trong IC 7402 hay 74LS02.

* Các IC cổng TTL thực hiện hàm NOT, AND, OR cho trên hình 2.21.

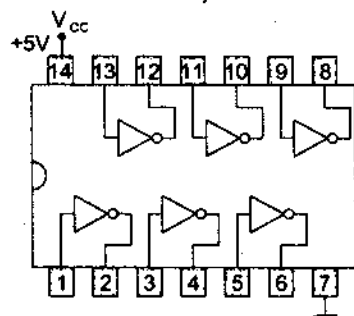
* TTL là viết tắt của cụm từ Transistor – Transistor – Logic thể hiện hình thức cấu trúc mạch điện như đã xét ở tiết trước, trong đó các đầu vào và đầu ra của IC số đều dùng BJT. Căn cứ vào số lượng cấu kiện được gia công trên một diện tích xác định, IC số được chia thành loại nhỏ (SSI) có chứa khoảng hơn một chục cổng, loại vừa (MSI) thường chứa hơn một trăm cổng, loại lớn (LSI) thường chứa không quá vài ngàn cổng và loại cực lớn (VLSI) có chứa tới hàng chục ngàn cổng trong IC. Trong đó 2 loại SSI và MSI là phổ biến nhất vì loại cổng TTL tiêu hao công suất đáng kể. Khi chế tạo BJT làm nhiệm vụ đóng ngắt trong cổng TTL, giải pháp chống bão hoà cho BJT nhờ mạch ghim mức ra dùng diốt rào thế Schottky có cấu tạo là một tiếp xúc phi tuyến giữa một kim loại và một chất bán dẫn tạp chất, diốt Schottky có điện áp mở thấp cỡ 0,4V, 0,5V nối song song với phần bazơ - collector của BJT ghim điện áp mở của BJT ở mức này,



a)



b)



c)

Hình 2.21

Cổng IC thực hiện các hàm

a) OR SN7432

b) AND SN7408

c) NOT SN7404

Cấu trúc và sơ đồ bố trí chân.

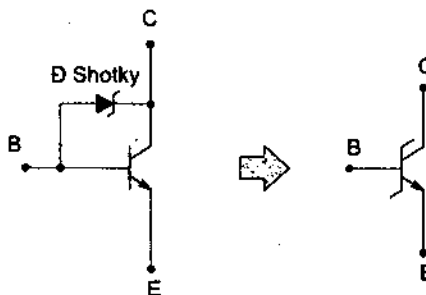
Yêu cầu mức cao $+2V \leq V_H \leq +5V$

Yêu cầu mức thấp $0V \leq V_L \leq 0,8V$.

nhờ đó BJT không rơi vào bão hoà và tốc độ lật trạng thái của BJT nâng lên từ vài lần đến vài chục lần. Hình 2.22 thể hiện ký hiệu quy ước của kết cấu BJT - diốt Shotky :

Thế hệ các IC cổng có cấu tạo như trên đối với BJT ở trong mạch được ký hiệu thêm chữ LS, ví dụ 74LS00 như đã gặp ở trên.

* Ưu điểm chủ yếu của IC hệ TTL là tốc độ đóng ngắt cao, điện áp ra (mức cao) đủ lớn, khả năng chịu tải và khả năng chống nhiễu khá tốt, tuy nhiên do mức tiêu thụ năng lượng lớn nên khả năng thích hợp để tạo nên loại LSI hay VLSI đặc biệt bị hạn chế. Khi sử dụng chúng cần quan tâm tới đặc tính truyền đạt điện áp (đặc tính chuyển trạng thái giữa mức cao và mức thấp).



Hình 2.22. BJT có diốt Shotky chống bão hoà.

* Các IC số dạng bộ đảo được sử dụng rộng rãi, đặc biệt khi cần biến đổi một số nhị phân thành số bù 1 của nó. Ví dụ với các cổng của SN7404 chứa 6 phân tử NOT, khi đưa tới các đầu vào chân 1,3,5,9,11,13 một số nhị phân 6 bit dạng : $Z = 100110$ (đồng thời nối chân số 7 với 0V chân 14 tới +5V), tại các đầu ra chân (chân số 2,4,6,8,10,12) ta sẽ nhận được số bù nhị phân (bù 1) của Z đã cho $\bar{Z} = 011001$. Với việc sử dụng 4 IC loại này đồng thời, tác động song song tới các đầu vào ta đã lập được số bù một của một số nhị phân gồm 3 byte.

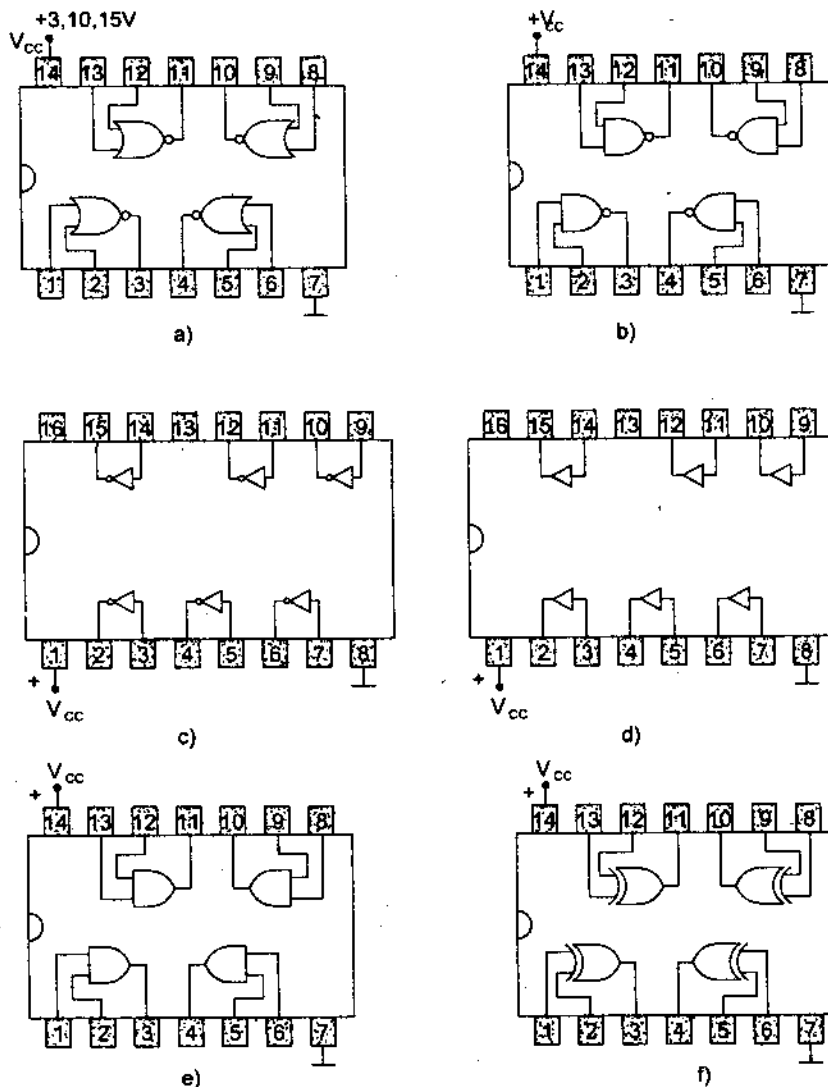
2.8.2. Các IC số loại CMOS thực hiện các hàm logic cơ bản

Các IC số loại CMOS sử dụng nguồn nuôi từ +3V tới +15V, có trở kháng đầu vào lớn do tính chất của tranzitor MOS, có bố trí 14 hoặc 16 chân ra. Điều đặc biệt cần lưu ý là các chân tín hiệu nếu không sử dụng đến cần phải được nối 0V vào chân số 7 với loại 14 chân hay 8 với loại 16 chân hoặc chân số 14 hay chân số 1 với loại 16 chân (+V_{cc}) để tránh hiệu ứng đánh thủng tĩnh điện.

Hình 2.23 cho cấu trúc khối bên trong của IC MOS cũng như cách bố trí chân tương ứng đối với các cổng cơ bản : hình 2.23a và b là nhóm IC chứa các cổng NOR(4001) và NAND(4011), mỗi IC gồm 4 cổng tương ứng. Hình 2.23c và d là các cổng đảo 4049 và các cổng đệm 4050.

* Ưu điểm chung của IC số họ CMOS là tiêu hao công suất nhỏ, có tốc độ chuyển mạch tương đối cao với biên độ ra (mức cao) lớn, phạm vi sử dụng nguồn điện áp rộng (từ +3V tới +18V), có hệ số tải đầu ra lớn và mức tạp âm cho phép lớn; nhược điểm chính là công nghệ chế tạo phức tạp và dễ hư hỏng vì đánh thủng tĩnh điện giữa cực G và kênh dẫn do vậy khi sử dụng cần chú ý nối 0V hoặc +V_{cc} các chân còn thừa (không dùng tới) để tránh hư hỏng. Một số mạch ứng dụng điển hình của các cổng IC TTL hay CMOS sẽ được đề cập ở tiết sau.

* Khi sử dụng nguồn +V_{cc} = 5V yêu cầu mức logic cao của IC MOS là $+3,5V \leq V_H \leq +5V$ còn yêu cầu mức logic thấp là $0 \leq V_L \leq +1,5V$.



Hình 2.23. IC họ CMOS cách bố trí chân và các cổng logic tương ứng và cổng đệm 4050 IC có 16 chân, chứa 6 cổng tương ứng
a) IC 4001 chứa 4 cổng NOR ; b) IC 4011 chứa 4 cổng NAND
c) IC4049 chứa 6 cổng NOT ; d) IC4050 chứa 6 cổng đệm (YES)
e) IC4081 chứa 4 cổng AND ; f) IC4070 chứa 4 cổng EXCLUSIVE-OR .

2.9. CÁC KÝ HIỆU LOGIC THAY THẾ

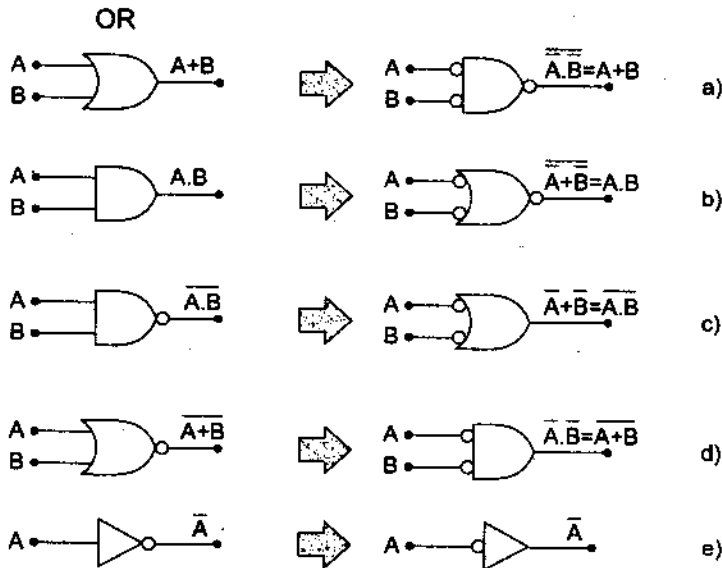
2.9.1. Các ký hiệu thay thế 5 ký hiệu chuẩn của các cổng cơ bản

Ở các tiết trên chúng ta đã tìm hiểu 5 loại cổng logic cơ bản nhất và các ký hiệu quy ước chuẩn tương ứng của chúng nhằm thể hiện chúng trên sơ đồ mạch logic. Ngày nay người ta sử dụng nhiều các sơ đồ có ký hiệu logic thay thế ngoài các ký hiệu chuẩn đã có. Hình 2.24 giới thiệu các ký hiệu thay thế 5 ký hiệu chuẩn, có được bằng cách:

* Đảo mỗi đầu vào và đầu ra của ký hiệu chuẩn đã có (thêm các vòng tròn nhỏ vào các đầu vào và đầu ra chưa có vòng tròn và loại bỏ các vòng tròn đã có sẵn ở ký hiệu chuẩn).

* Thay đổi ký hiệu đại số OR $\leftrightarrow$ AND

Với bộ đảo ký hiệu cổng không thay đổi.



Hình 2.24. Các ký hiệu logic thay thế tương đương các ký hiệu chuẩn
(ký hiệu logic tương đương)

Chú ý :

– Trong bộ ký hiệu chuẩn không có ký hiệu nào có vòng tròn nhỏ (đảo) ở đầu vào trong khi ở các ký hiệu tương đương thay thế đều có vòng tròn.

– Về mặt vật lý không có sự khác nhau của các mạch biểu diễn chúng, ở đây bản chất hàm logic và biểu diễn vật lý hoàn toàn giống nhau.

– Ký hiệu logic thay thế tương đương có thể mở rộng cho các cổng có số đầu vào lớn hơn 2.

– Với cổng OR và AND là các cổng không đảo nên ký hiệu thay thế của chúng xuất hiện vòng tròn đảo ở cả đầu vào và đầu ra.

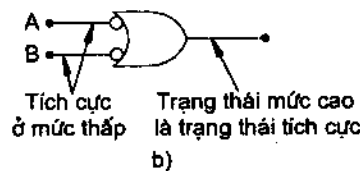
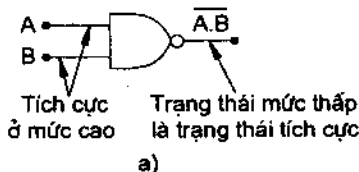
Với cổng NOR và NAND chúng là các cổng có đảo nên cả hai loại ký hiệu chuẩn và ký hiệu tương đương đều có vòng tròn đảo ở đầu vào hoặc đầu ra.

2.9.2. Ý nghĩa của các ký hiệu tương đương thay thế

* Khái niệm về mức logic tích cực: Trên ký hiệu mạch logic, khi một đường vào hay đường ra nào đó không có vòng tròn nhỏ đường đó được gọi là tích cực ở mức cao. Còn khi nếu trên các đường này có vòng tròn nhỏ, ta gọi đường đó là tích cực ở mức thấp. Nghĩa là sự có mặt của vòng tròn nhỏ trên 1 đường nối tín hiệu sẽ quyết định trạng thái tích cực của đường này là ở mức thấp (khi có vòng tròn nhỏ) hay là ở mức cao (khi không có vòng tròn nhỏ). Qua khái niệm mức logic tích cực có thể giải thích hoạt động của mạch rõ ràng hơn.

* Hãy lấy ký hiệu cổng NAND làm ví dụ (hình 2.25a,b)

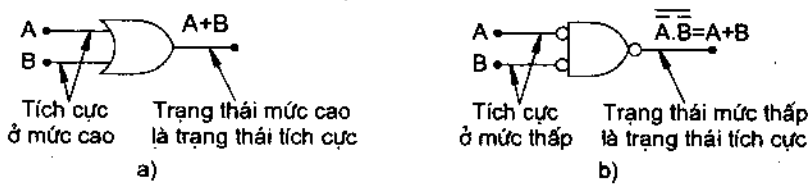
Qua ví dụ đã nêu ta thấy rõ cách ký hiệu đã mô tả hai mệnh đề trạng thái đối lập nhau của một đối tượng. Vấn đề là trạng thái nào đang được quan tâm, với 2.25a điều quan tâm là trạng thái mức thấp của tín hiệu ra, còn ở 2.25b lại quan tâm chú ý tới trạng thái mức cao của tín hiệu ra.



Hình 2.25. Giải thích ý nghĩa hai cách ký hiệu cổng NAND

- a) Kết luận từ ký hiệu này là : " đầu ra chỉ xuống mức thấp khi mọi đầu vào đều ở mức cao" (phép logic AND) ;
b) Kết luận từ ký hiệu này là : " đầu ra sẽ lên mức cao khi có ít nhất một đầu vào bất kỳ ở mức thấp" (phép logic OR) .

* Đối với cổng OR hai cách ký hiệu và giải thích ý nghĩa của chúng được cho trên hình 2.26.



Hình 2.26. Giải thích ý nghĩa hai cách ký hiệu cổng OR

- a) Kết luận từ ký hiệu này là : Đầu ra sẽ lên mức cao khi có ít nhất một đầu vào bất kỳ ở mức cao ;
b) Kết luận từ ký hiệu thay thế là : Đầu ra chỉ xuống mức thấp khi mọi đầu vào đều ở mức thấp.

2.9.3. Cách sử dụng hai dạng ký hiệu chuẩn và ký hiệu thay thế

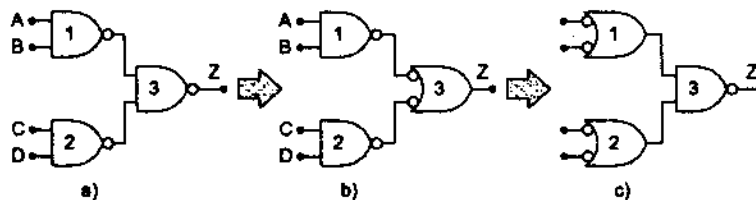
Một câu hỏi đặt ra cho người thiết kế mạch là nên sử dụng loại ký hiệu nào ? Điều này xuất phát từ đầu ra tác động tới mạch chấp hành (tải) như thế nào ?

* Nếu sử dụng trạng thái ra Z để gây nên một hành động nào đó (kích hoạt động chuông báo động bằng role, mở đèn báo sáng...) yêu cầu $Z = 1$ khi đó ta cần Z tích cực ở mức cao.

* Ngược lại nếu cần sử dụng trạng thái của Z ở mức thấp ta sẽ biểu diễn theo cấu trúc có Z tích cực ở mức thấp.

Khi thực hiện thay thế ký hiệu cần chú ý quy tắc khi ghép liên tiếp các ký hiệu cổng thì đầu ra có vòng tròn nhỏ phải nối với đầu vào có vòng tròn nhỏ và ngược lại đầu ra của cổng trước không có vòng tròn nối tới đầu vào của cổng sau cũng không có vòng tròn (để đảm bảo tính chất tương thích giữa các mức logic tích cực khi truyền tín hiệu từ cổng này tới cổng kia). Khi sử dụng quy tắc này sẽ xuất hiện các sơ đồ logic thực tế dùng cả hai loại ký hiệu chuẩn và ký hiệu thay thế trong cùng một cấu trúc cụ thể.

Ví dụ 1 : Sơ đồ chuẩn tạo hàm Z có dạng hình 2.27a. Sau khi chỉ thay thế cổng 3 ta có mạch 2.27b và nếu thay thế 2 cổng NAND 1 và 2 ta có mạch 2.27c. Cấu trúc 2.27b thể hiện yêu cầu trạng thái ra ở mức tích cực cao và cấu trúc 2.27c yêu cầu mức thấp của Z là mức logic tích cực. Các đường nối trung gian giữa các cổng 1,2 và cổng 3 phù hợp trạng thái tích cực với nhau. Mạch 2.27b thể hiện “Z sẽ lên mức cao bất cứ khi nào $A = B = 1$ hoặc $C = D = 1$ ”



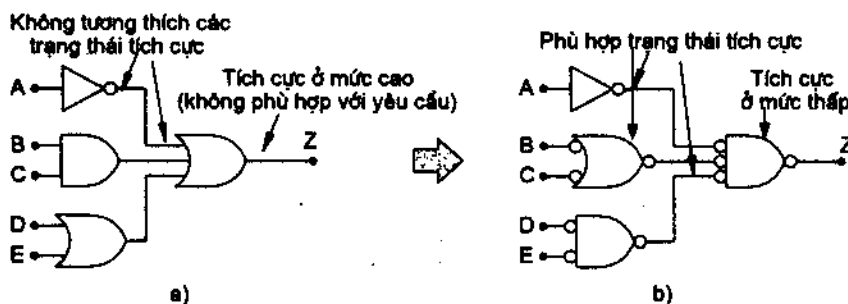
Hình 2.27. Ba cách biểu diễn cấu trúc thực hiện hàm Z cho ở ví dụ 1

còn cấu trúc 2.27c thể hiện “Z sẽ xuống mức thấp chỉ khi A hoặc B và C hoặc D ở mức thấp” thể hiện rõ trong bảng chân lý của Z (bảng 2.2).

Bảng 2.2. CHO VÍ DỤ 1

Biến	A	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	B	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
	C	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
	D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Z		0	0	0	1	0	0	0	1	0	0	0	1	1	1	1

Ví dụ 2 : Mạch hình 2.28a sử dụng để kích hoạt động một cổng logic khác khi đầu ra Z xuống mức thấp. Hãy hiệu chỉnh sơ đồ logic để mô tả hoạt động của mạch có hiệu quả hơn.



Hình 2.28

a) Mạch ban đầu có hai nhược điểm ; b) Mạch đã hiệu chỉnh.

Mạch hình 2.28b sau khi đã hiệu chỉnh có 2 ưu điểm quan trọng:

- . Trạng thái ra là tích cực ở mức thấp theo yêu cầu của bài toán.
- . Các đường phối ghép tín hiệu giữa các cổng 1,2,3 và 4 là tương thích trạng thái tích cực thấp với nhau.

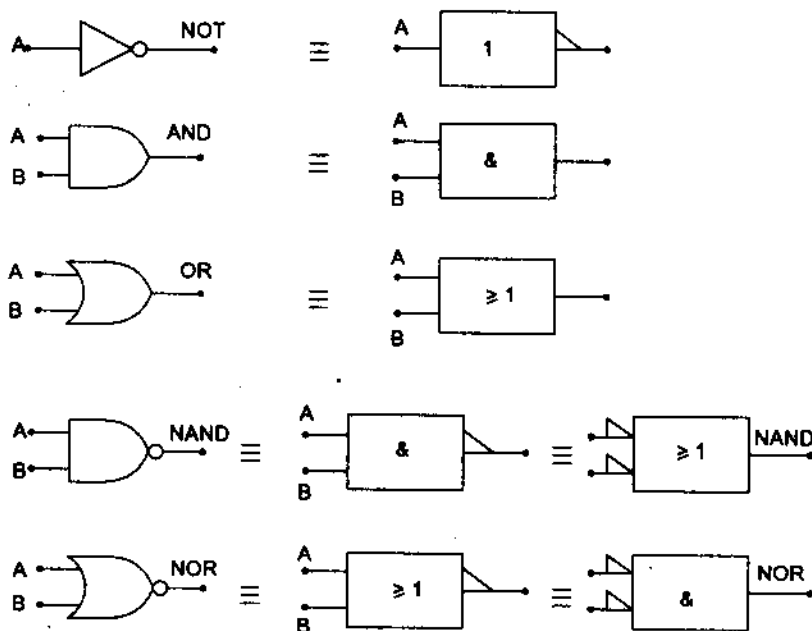
2.9.4. Ký hiệu các cổng logic theo tiêu chuẩn IEEE/ ANSI

Để cung cấp đầy đủ thông tin về các cổng logic đối với các hệ thống logic phức tạp, các ký hiệu cổng chuẩn hay cổng tương đương thay thế tuy rõ ràng và đầy đủ đối với các mạch đơn giản có độ phức tạp không cao nhưng sẽ trở nên cồng kềnh và không đủ thông tin về nó khi mạch trở nên phức tạp.

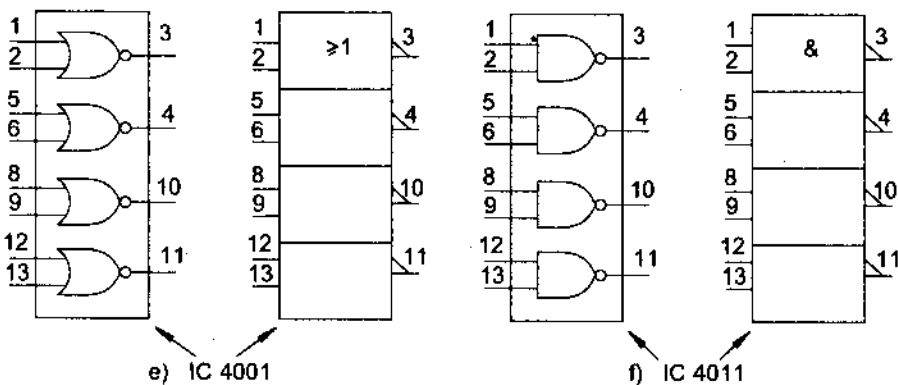
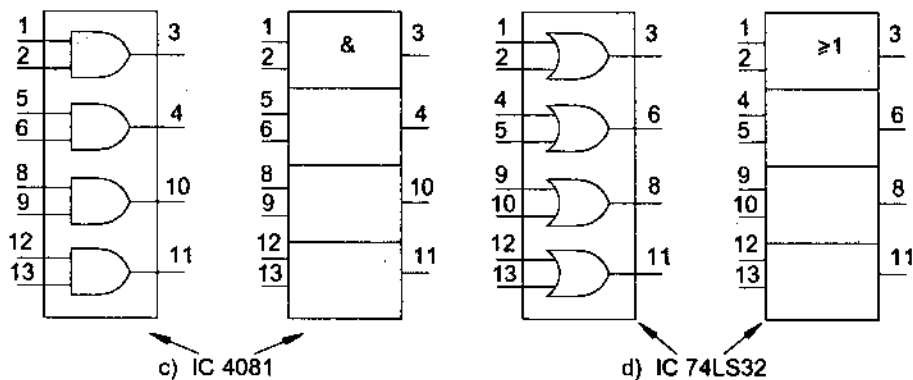
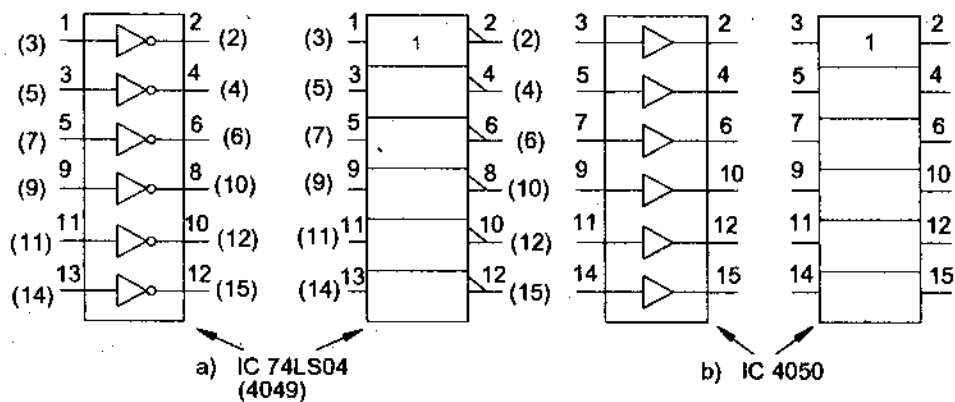
Vì lý do trên các ký hiệu theo chuẩn IEEE/ ANSI ra đời.

1. Điểm khác nhau căn bản là trong chuẩn IEEE/ ANSI dùng ký hiệu hình chữ nhật cho tất cả các cổng logic thay vì nhiều ký hiệu khác nhau cho mỗi loại cổng, kèm theo đó là một hệ thống ký pháp phụ thuộc được sử dụng để chỉ ra tính phụ thuộc của đầu ra đối với đầu vào.

2. Trên hình 2.29 chỉ ra các ký hiệu chữ nhật tương đương biểu diễn các cổng truyền thống.



Hình 2.29. Các ký hiệu chuẩn chữ nhật (IEEE/ ANSI) thay thế các ký hiệu cổng logic truyền thống.



Hình 2.30. Biểu diễn các cổng IC số theo ký hiệu chữ nhật IEEE/ ANSI.

3. Một số đặc điểm của ký hiệu chữ nhật cần chú ý là :

– Tam giác vuông nhỏ được thay thế vòng tròn nhỏ ở chỗ nó có biểu thị phép đảo hay đường tín hiệu với logic tích cực thấp ở đầu vào hoặc đầu ra của ký hiệu chữ nhật.

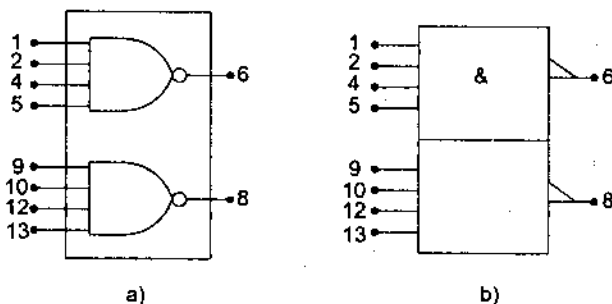
– Bên trong ký hiệu chữ nhật dùng một ký pháp đặc biệt mô tả mối quan hệ logic giữa đầu ra và đầu vào : Số “1” và tam giác vuông nhỏ ở đầu ra trong ký hiệu bộ đảo cho biết cổng chỉ có một đầu vào và đầu ra sẽ chuyển về mức tích cực thấp khi đầu vào ở mức tích cực cao. Dấu “&” thể hiện quan hệ hàm AND nghĩa là đầu ra về mức tích cực cao khi mọi đầu vào ở mức tích cực cao. Dấu “≥ 1” thể hiện quan hệ hàm OR : đầu ra đạt mức tích cực cao bất cứ khi nào có một hoặc từ một đầu vào trở lên ở vào trạng thái tích cực cao...

– Ký hiệu chữ nhật biểu diễn cổng NAND, NOR giống như của ký hiệu biểu diễn cổng AND, OR tương ứng có bổ sung thêm tam giác ở đầu ra thể hiện các trạng thái tích cực thấp đạt được khi các đầu vào thỏa mãn các điều kiện của phép logic tương ứng.

4. Với các IC cổng, ký hiệu chữ nhật biểu diễn đặc biệt thuận lợi, ví dụ các IC cổng trong hình 2.21 hay các IC cổng trong hình 2.23 được biểu diễn lại theo quy ước này trên hình 2.30.

Ta chú ý là trong ký hiệu chữ nhật IC cổng ví dụ 4001 có 4 cổng NOR độc lập nhau, khi đó chỉ cần ký pháp trong ô chữ nhật đầu tiên là đủ, các ô sau bỏ trống không cần ghi ký pháp “≥ 1” vào chúng nữa. Các ký hiệu chữ nhật với ký pháp quan hệ cho phép người sử dụng thông qua sơ đồ logic không cần tra cứu bảng IC có thể nhận biết ngay quan hệ logic nó thực hiện.

Ta lấy 1 ví dụ khác, với IC cổng NAND 4 cửa vào 74HC20 cho trên hình 2.31.



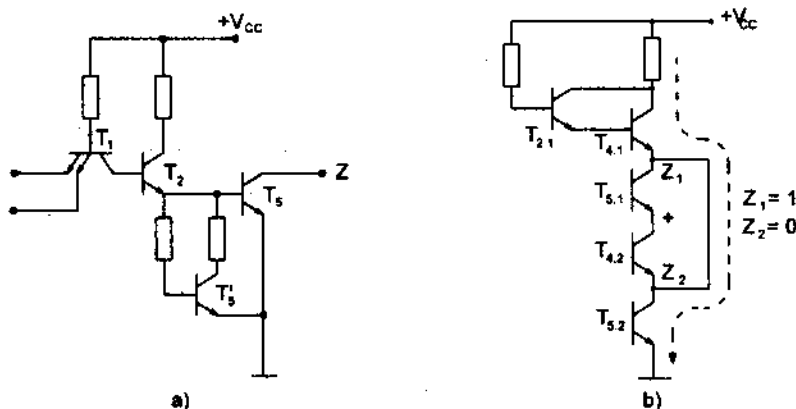
Hình 2.31

a) Cấu trúc IC cổng 74HC20 ; b) Sơ đồ ký hiệu thay thế kiểu chữ nhật.

2.10. CÁC MẠCH ĐIỆN CỔNG KHÁC

Ngoài các cổng cơ bản đã được xét ở trên, trong tiết này sẽ giới thiệu một số loại cổng thông dụng khác đáp ứng các yêu cầu hàm logic đa dạng và cần thiết trong thực tế. Tuy nhiên phần lớn các loại cổng này đều có cấu trúc cơ bản dựa trên cổng NAND (hình 2.10).

2.10.1. Cổng NAND hở Colector (hình 2.32)



Hình 2.32

- a) Cổng NAND hở colector (T_5 để giảm thời gian chuyển mạch của T_5)
 b) Mạch ra của hai cổng NAND hình (2.10) nối trực tiếp với nhau khi $Z_1=1$ và $Z_2=0$.

* Do đầu ra của cổng NAND TTL có điện trở ra thấp dù Z ở mức tích cực cao hay thấp nên không thể đấu hai cổng NAND song song với nhau (Z_1 với Z_2 như hình 2.32b) để tạo ra hiệu ứng của hàm AND được. Khi $Z_1=1$ và $Z_2=0$ sẽ xuất hiện 1 dòng điện lớn chảy qua T_{41} xuống T_{52} làm mức thấp cổng NAND₂ tăng lên và làm hỏng mức cao cổng NAND₁.

* Với mạch hình 2.32a để hở colector hiện tượng trên sẽ không xảy ra khi nối song song 2 cổng NAND loại này để tạo hàm AND ở cổng ra $Z_1.Z_2=Z$. Trong cổng loại này colector của T_5 để hở lơ lửng, khi sử dụng phải chọn điện trở $R_{tải}$ để mắc vào colector của T_5 (gọi tắt là cổng NAND OC – Open Collector). Việc lựa chọn điện trở $R_{tải}$ mắc vào colector của cổng phải quan tâm tới các tham số như:

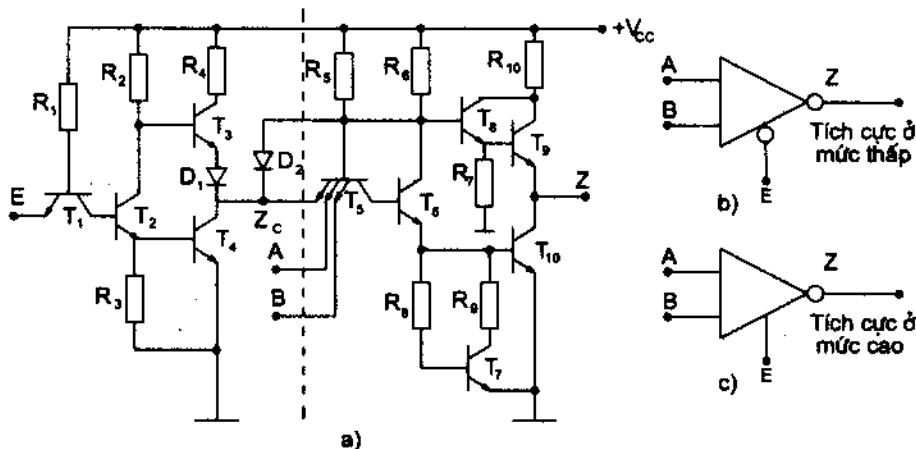
- Số lượng cổng loại này đấu song song ở cửa ra (ví dụ n cổng).

- Số lượng m cổng có đầu vào nối vào đầu ra song song này.
- Các mức điện thế nguồn nuôi, điện thế mức cao chuẩn, dòng điện dò qua T_5 lúc bị ngắt mạch và dòng điện vào của mỗi phụ tải đầu tới...

2.10.2. Cổng NAND có đầu ra 3 trạng thái (NAND-TS Three state)

Mạch nguyên lý của loại cổng NAND - TS cho trên hình 2.33.

- Đầu ra Z của cổng loại này có 3 trạng thái: mức cao, mức thấp, cấm (trạng thái có trở kháng cao). Phần bên phải đường đứt nét chính là cổng NAND₂ 2 cửa vào A và B hình 2.10 đã xét (ở đây có bổ sung T_7 cải tiến quá trình phóng điện cho



Hình 2.33. a) Cổng NAND - TS ; b, c) Ký hiệu quy ước.

T_{10} lúc chuyển từ nối sang ngắt và kích mở nhanh cho nó khi chuyển từ ngắt sang nối mạch khi T_6 được nối mạch). Phần bên trái cũng là một cổng NAND₁ có 1 cửa vào E có cải tiến một chút như T_3 không mắc Darlington, có thêm D_1 để đảm bảo việc ngắt tin cậy của T_3 lúc T_2 và T_4 thông nối mạch.

Khi $E = 0$ (ở mức thấp), tại cửa ra của NAND₁ có mức cao $Z_C = 1 = \overline{E}$ lúc đó cho phép NAND₂ làm việc với 2 đầu vào A và B thực hiện chức năng của một cổng NAND bình thường: $Z = \overline{A.B}$.

- Khi $E = 1$ (ở mức cao), $Z_C = 0$ làm T_6, T_7, T_{10} ngắt mạch, (T_5 thông nối mạch theo emitter tương ứng với cổng Z_C). Mức $Z_C = 0$ qua D_2 nối mạch làm ngắt

mạch T_8, T_9 (mức thế bazơ V_{B8} của T_8 bị ghim ở mức xấp xỉ 1V). Từ đầu ra nhìn vào, mạch điện ở trạng thái trở kháng cao (bị treo không có liên hệ với các cổng vào nối vào phía sau nó hay với một đường truyền tín hiệu nối ở đầu ra của nó). Kết quả chung hàm ra Z có dạng $Z = \overline{A.B.E}$.

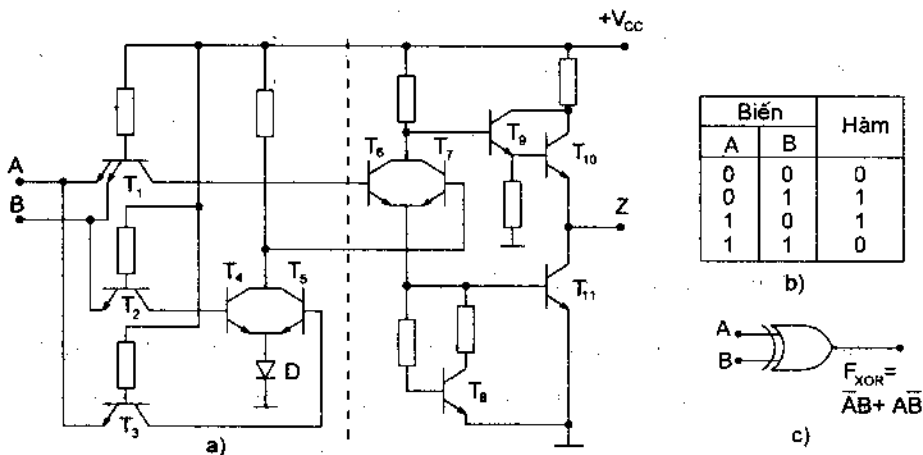
– Ký hiệu quy ước cổng NAND – TS cho trên hình 2.33b, ngoài ra ta còn gặp loại quy ước mức cấm là tích cực cao (hình 2.33c).

– Các cổng IC có 3 trạng thái cho phép nối song song các đầu ra với nhau mà không làm giảm tốc độ chuyển mạch với yêu cầu luôn chỉ có một cổng được phép hoạt động theo sự điều khiển của đầu vào E.

2.10.3. Cổng hoặc tuyệt đối (XOR) thực hiện quan hệ logic

$$\begin{aligned} F_{XOR} &= \overline{A}B + A\overline{B} \\ &= A \oplus B \end{aligned} \quad (2.7)$$

Mạch điện nguyên lý một cổng XOR TTL được cho trên hình 2.34a, bảng trạng thái và ký hiệu quy ước cổng XOR cho trên hình 2.34b và c.



Hình 2.34. a) Mạch điện nguyên lý cổng XOR (EXCLUSIVE- OR)
b) Bảng trạng thái ; c) Ký hiệu quy ước.

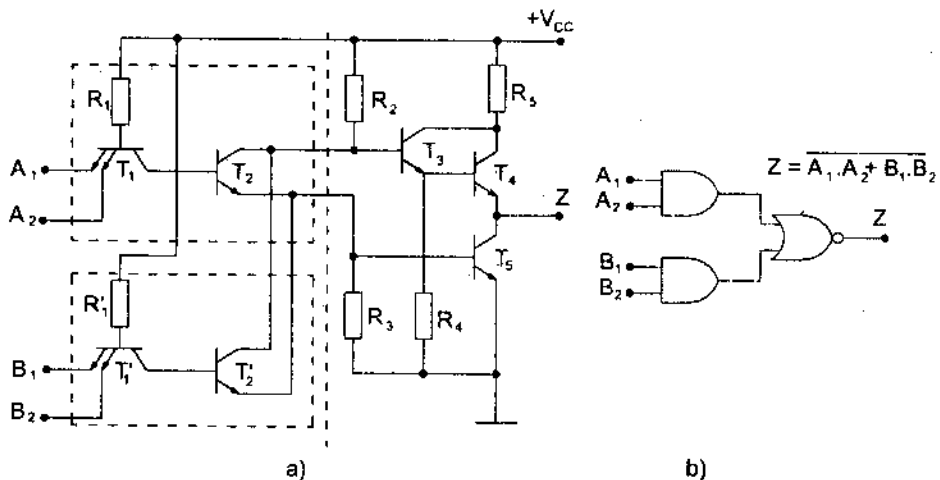
– Trong mạch điện hình 2.34a phần phải đường đứt nét cấu tạo tương tự như mạch điện tầng ra và tầng giữa của cổng NAND, chỉ khác là T_6 và T_7 đấu song song nhau (giữa collector và emitter của chúng) do vậy, cũng như nguyên lý hoạt động của cổng NAND hình 2.10a, khi một trong chúng nối mạch (thông báo hoà) thì T_9 ,

T_{10} ngắt, T_{11} nối và $Z = 0$ (đầu ra ở mức thấp). Khi $A = B = 1$ (mức cao) T_1 ngắt, T_6 và T_{11} nối mạch do vậy $Z = 0$. Khi $A = B = 0$ (mức thấp) T_1, T_2 và T_3 nối mạch làm cặp T_4, T_5 ngắt mạch T_7 và T_{11} nối mạch do đó đầu ra ở mức thấp $Z = 0$. Khi có một trong các đầu vào ở mức cao (ví dụ $A = 1, B = 0$) T_1 nối mạch, T_6 ngắt mạch, T_3 ngắt làm T_5 nối mạch làm V_{B7} có điện thế thấp cỡ 1V, $V_{E4} = V_{E5} = V_D$ của diôt, T_7 ngắt đẩy T_{11} ngắt mạch và $Z = 1$. Các trạng thái của Z vừa phân tích tuân theo quy luật của bảng 2.34b.

– Hàm F_{XOR} sử dụng phổ biến trong các mạch thực tế, IC cổng 4070 (hình 2.23f) chứa 4 cổng XOR loại CMOS. Phép tính thực hiện theo biểu thức 2.7 chính là phép cộng nhị phân giữa A và B (không nhớ). Đầu ra sẽ ở mức thấp ($Z = 0$) khi giá trị các biến logic A và B ở cổng vào giống nhau.

2.10.4. Cổng NORAND (hình 2.35)

– So sánh với mạch (2.10) của cổng NAND, ở đây có thêm T_1', R_1', T_2' đấu song song với T_1, R_1, T_2 (T_2 và T_2' nối song song nhau ở đầu ra phần trái đường đứt nét). Khi có bất kỳ một trong hai tranzitor T_2 hoặc T_2' nối mạch (thông báo hoà) sẽ làm cho T_5 nối mạch, T_4 ngắt mạch, đầu ra ở mức thấp ($Z = 0$). Chỉ khi T_2 và T_2' đồng thời ngắt mạch T_4 nối và T_5 ngắt, đầu ra chuyển lên mức cao ($Z = 1$).



Hình 2.35. a) Mạch điện cổng logic NORAND (và - hoặc - đảo)
b) Ký hiệu cổng NORAND.

– Vậy quan hệ logic ở đây là khi nhóm đầu vào A_1, A_2 hoặc B_1, B_2 đều ở mức cao thì $Z = 0$, chỉ cần trong 1 nhóm đầu vào nào đó có một giá trị thấp thì $Z = 1$, ta có quan hệ:

$$F_{\text{NORAND}} = \overline{A_1 \cdot A_2 + B_1 \cdot B_2} \quad (2.8)$$

Như vậy trong mỗi nhóm đầu vào T_1 hoặc T_1' thực hiện quan hệ AND, do T_2 và T_2' đầu song song nên thực hiện quan hệ OR giữa 2 nhóm A_i và B_i . Mạch tăng ra thực hiện quan hệ NOT.

– Cổng NORAND có ký hiệu quy ước cho trên hình 2.35b.

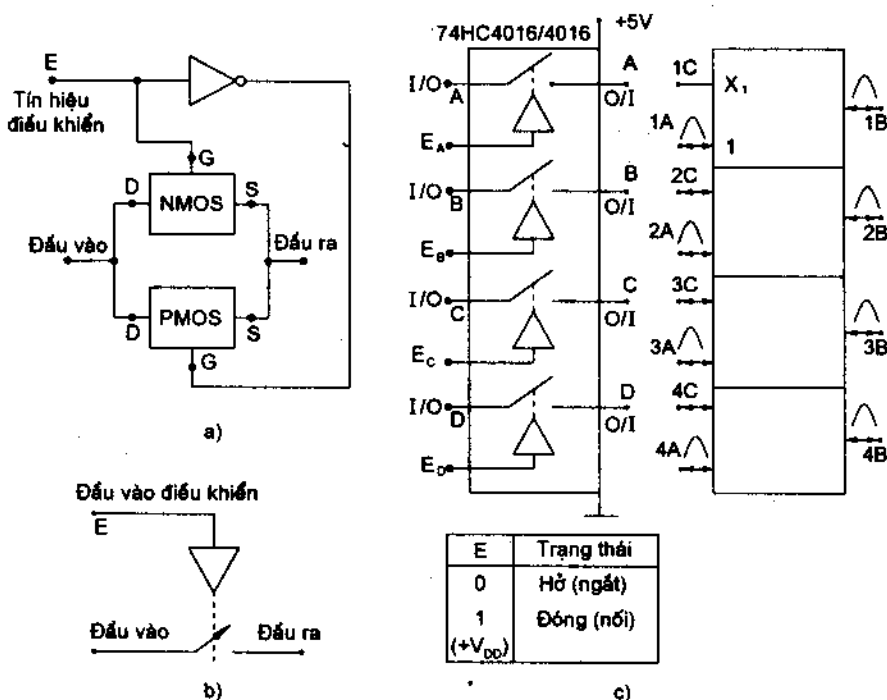
2.10.5. Cổng truyền CMOS

Mạch điện cấu tạo của một cổng truyền CMOS (hình 2.36a), ký hiệu quy ước chuẩn (hình 2.36b) và ký hiệu chữ nhật trên hình 2.36c cho IC 4016/74HC4016 (gồm 4 cổng độc lập).

– Mạch 2.36a hoạt động như một chuyển mạch đơn cực một vị trí được điều khiển bằng mức logic đầu vào, có khả năng truyền tín hiệu theo cả hai hướng. Mạch gồm một cặp NMOS và PMOS (sinh đôi) đầu song song nhau chịu sự điều khiển của tín hiệu điều khiển E. Khi $E = 1$ (mức cao) cả 2 MOS được nối mạch và chuyển mạch ở trạng thái đóng (nối mạch giữa đầu vào và đầu ra), khi $E = 0$ (mức thấp) cả 2 tranzitor MOS ngắt mạch tạo trạng thái hở mạch giữa đầu vào và đầu ra.

Trong ký hiệu chữ nhật ở hình 2.36c mỗi IC 4016/74HC4016 chứa 4 cổng chuyển mạch CMOS hình (2.36a), tín hiệu điều khiển E được dán mác X_1 trong đó X thể hiện năng lực điều khiển của nó tới hoạt động truyền tín hiệu 2 chiều (ký pháp “ $\leftrightarrow$ ”) tại các đầu vào và đầu ra của cổng có chỉ số 1 của nhãn cổng. Dấu ký pháp “ $\cap$ ” thể hiện các đầu vào ra này không chỉ truyền các tín hiệu số mà còn truyền được cả tín hiệu tương tự (tín hiệu liên tục theo thời gian).

– Khi sử dụng các cổng của IC 4316/74HC4316 cũng có 4 chuyển mạch 2 chiều nuôi bằng nguồn $+V_{DD}$ và $-V_{EE}$, như vậy mức cao của nó là $+V_{DD}$ và mức thấp là $-V_{EE} < 0$ và cho phép tín hiệu đầu vào biến thiên từ $-V_{EE}$ tới $+V_{DD}$ (tín hiệu lưỡng cực, tương tự hay số).



Hình 2.36. Cổng chuyển mạch CMOS
a) Cấu tạo mạch điện (khối); b) Ký hiệu quy ước và bảng trạng thái;
c) Ký hiệu IC 4016 dạng chuẩn và dạng chữ nhật.

2.11. GHÉP NỐI CÁC CỔNG IC SỐ

2.11.1. Các thuật ngữ tham số của IC số

Trước khi sử dụng các IC số hoặc thực hiện phối ghép chúng, người sử dụng cần quan tâm và nắm vững các tham số mô tả tính chất của IC số, bên cạnh những tham số tính chất đã nói tới ở tiết 2.1, cần bổ sung chi tiết một số tham số quan trọng nhất:

1. Tham số dòng điện và điện thế

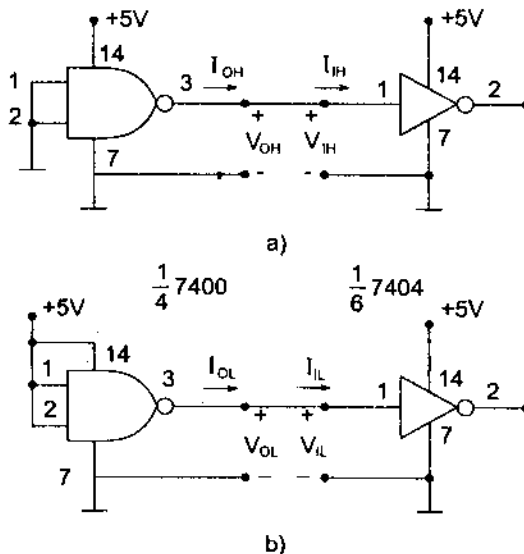
– Mức điện thế tối thiểu để có logic 1 tại đầu vào: V_{IHmin} hay tại đầu ra: V_{OHmin} ; mọi giá trị điện thế thấp hơn mức này cổng logic sẽ không chấp nhận là mức cao tại các đầu vào, ra tương ứng.

– Mức điện thế tối đa để còn là logic 0 tại đầu vào : V_{ILmax} hay tại đầu ra : V_{OLmax} ; mọi giá trị điện thế cao hơn mức này cổng logic sẽ không chấp nhận là mức thấp tại các đầu vào và ra tương ứng.

(Với các mức V_{OHmin} và V_{OLmax} cần xét dưới các điều kiện tải xác định)

Trong các cổng TTL người ta quy định $V_{IHmin} = +2V$; $V_{OHmin} = +2,4V$ đến $+2,7V$; $V_{ILmax} = +0,8V$; $V_{OLmax} = +0,4V$ đến $+0,5V$, còn đối với các cổng CMOS các mức này tương ứng là $V_{IHmin} = +2V$ đến $+3,5V$, $V_{OHmin} = +4,4V$ đến $+4,9V$; $V_{ILmax} = +1,5V$; $V_{OLmax} = 0,1V$ (xem bảng 2.4).

– Dòng điện tối đa tại cổng vào mức cao I_{IHmax} hay tại cổng ra mức cao I_{OHmax} .



Hình 2.37. Các tham số điện áp và dòng điện tại đầu ra, đầu vào khi ghép 2 cổng TTL
a) Trạng thái mức cao (logic 1) ; b) Trạng thái mức thấp (logic 0).

– Dòng điện tối đa tại cổng vào mức thấp I_{ILmax} hay tại cổng ra mức thấp I_{OLmax} (tại cổng vào xác định theo mức thế tương ứng, tại cổng ra xác định theo tải chỉ định) (xem bảng 2.3). Hình 2.37 minh họa các tham số dòng và thế vừa nêu trên khi ghép 2 cổng TTL 7400 với 7404.

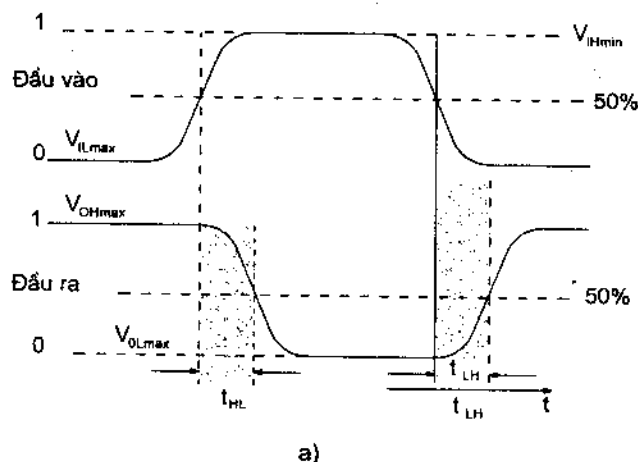
2. Hệ số tải được định nghĩa là số lượng đầu vào logic chuẩn tối đa mà một đầu ra (phía trước) có thể kích thích một cách đáng tin cậy, nếu vượt quá số lượng này sẽ không đảm bảo giá trị điện thế mức logic tại đầu ra.

3. Thời gian trễ do truyền tín hiệu qua cổng gồm 2 phần:

– Thời gian trễ lúc chuyển từ mức 0 lên mức 1 t_{LH} (tính khi mức đạt tới 50%)

– Thời gian trễ lúc chuyển từ mức 1 về mức 0 t_{HL} (tính khi mức đạt tới 50%)

Ví dụ : Khi truyền qua 1 bộ đảo, xung vuông góc bị trễ thể hiện qua hình 2.38a



Hình 2.38a. Thời gian trễ khi tín hiệu truyền qua một cổng đảo $t_{tr} = t_{HL} + t_{LH}$

4. Công suất tiêu hao trung bình của cổng khi cổng được nuôi bởi $+V_{cc}$

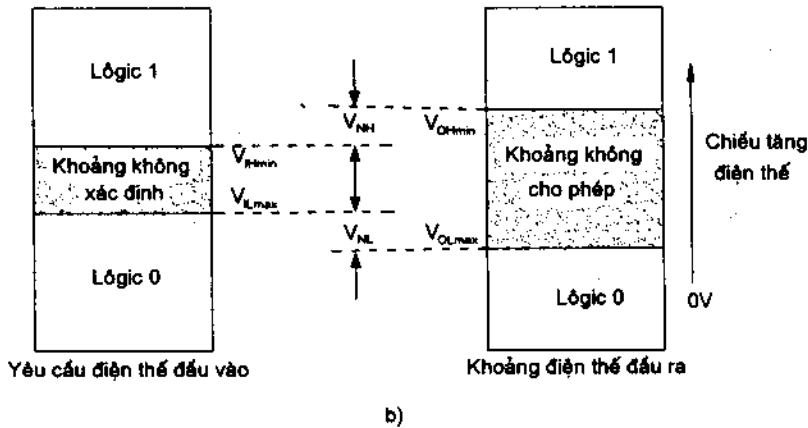
$$P_o = V_{cc} \cdot I_o$$

$$\text{ở đây } I_o = \frac{I_{ccL} + I_{ccH}}{2}$$

Với cổng NAND: I_{ccH} là dòng tiêu thụ của IC khi tất cả các đầu ra ở mức cao. I_{ccL} là dòng tiêu thụ của IC khi tất cả các đầu ra của nó ở mức thấp (coi thời gian các cổng có đầu ra ở mức thấp và mức cao là xấp xỉ nhau). Người ta còn đưa ra tham số tích số (tốc độ $\times$ công suất) để đánh giá hiệu suất chung của IC số. Tốc độ đánh giá bằng thời gian trễ khi truyền tín hiệu qua cổng.

5. Khả năng chống nhiễu được xác định như khả năng chịu được các tác động của tín hiệu giả do điện từ trường lạ gây ra mà không làm thay đổi mức điện thế ở đầu ra. Số đo định lượng khả năng chống nhiễu gọi là giới hạn nhiễu.

Hình 2.38b mô tả cách xác định các giới hạn nhiễu mức cao V_{NH} và mức thấp V_{NL} tương ứng cho các cửa vào và cửa ra của một cổng logic.



Hình 2.38b. Xác định giới hạn nhiễu

Từ hình 2.38b thấy rõ việc xác định giới hạn nhiễu:

- Ở trạng thái cao $V_{NH} = V_{OHmin} - V_{IHmin}$
- Ở trạng thái thấp $V_{NL} = V_{ILmax} - V_{OLmax}$

mọi trạng thái ở cổng vào có giá trị trong khoảng $V_{ILmax} < V_I < V_{IHmin}$ đều có thể gây ra phản ứng tại cổng ra ngoài dự kiến (sai) và gọi là các mức vào không hợp lệ.

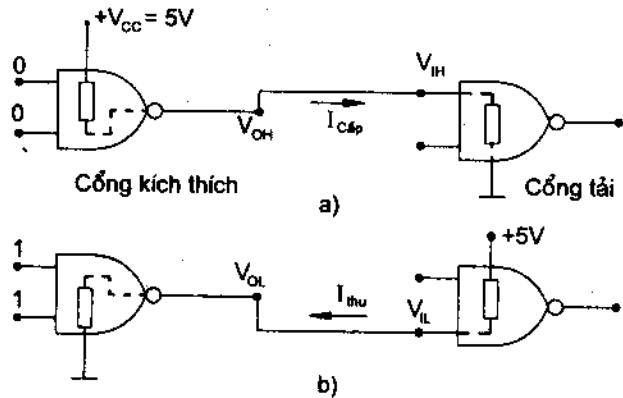
2.11.2. Hai phương thức hoạt động khi ghép nối các cổng logic với nhau xét theo cách truyền tải dòng điện

1. Cổng logic trước cung cấp dòng cho cổng sau: khi đầu ra của cổng trước ở logic 1 nó sẽ cung cấp dòng (fun dòng) tới đầu vào cổng sau nối liền tiếp với nó (vai trò như một điện trở tải nối tới 0V).

2. Cổng logic trước thu nhận dòng từ cổng sau : khi đầu ra của cổng trước ở mức thấp (logic 0) nó sẽ kích thích đầu vào cổng sau truyền tải dòng ngược về cổng trước qua điện trở ra của cổng trước. Các quá trình này được minh họa trên hình 2.39 cho các cổng NAND ghép liên tiếp nhau.

– Đối chiếu với mạch điện cổng NAND hình 2.10 trường hợp cấp dòng : dòng điện từ V_{cc} qua R_5 qua T_4 nối mạch qua cửa Z lối ra (mức 1) vào T_1 của cổng tải. Còn trong trường hợp thu dòng: dòng điện từ $+V_{cc}$ qua R_1 qua T_1 của cổng tải tới đầu ra Z của cổng kích thích qua T_5 xuống mạch đất. Khi cổng trước của ra có logic 0 ($Z = 0$).

- Khi cổng tải có nhiều đầu vào nối tới một đầu ra của cổng kích thích phương thức truyền dòng cũng tương tự, từ đó hệ số mắc tải của cổng kích thích do khả năng chịu dòng điện của T_4 lúc cấp dòng và của T_5 lúc hút nhận dòng trong mạch điện tăng ra của cổng NAND đã xét trên mạch cơ bản hình 2.10.



Hình 2.39

a) Phương thức cấp dòng; b) Phương thức thu dòng

Ví dụ loại cổng NAND 7400 có các tham số:

$$I_{OL\max} = 16\text{mA}; I_{IL\max} = 1,6\text{mA}; I_{OH\max} = 0,4\text{mA}; I_{IH\max} = 40\mu\text{A}$$

$$\text{Từ đây hệ số tải mức thấp là } n = \frac{16\text{mA}}{1,6\text{mA}} = 10$$

Tức là 1 cổng kích thích T_5 có khả năng nhận dòng từ 10 cổng vào tải ở chế độ thu dòng. Hệ số tải ở mức cao là: $\frac{0,4\text{mA}}{40\mu\text{A}} = 10$ số lượng đầu vào của các cổng tải có thể được kích thích ở trạng thái cao là 10.

3. Phương pháp xác định hệ số tải cho các cổng bất kỳ khi ghép nối chúng

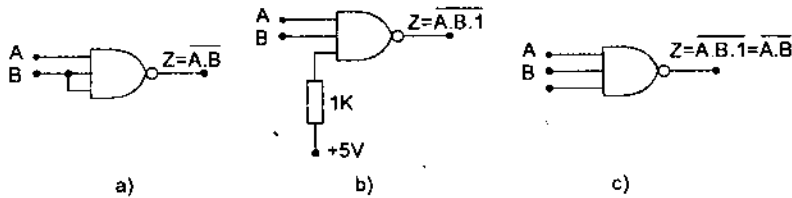
- Cộng góp dòng vào mức cao I_{IH} của tất cả các đầu vào tải được nối đến một đầu ra. Yêu cầu $\Sigma I_{IH} < I_{OH}$ của đầu ra của cổng kích thích.

- Cộng góp dòng vào mức thấp I_{IL} của tất cả các đầu vào nối đến một đầu ra. Yêu cầu $\Sigma I_{IL} < I_{OL}$ của đầu ra.

2.11.3. Một số đặc điểm của logic họ TTL và logic họ CMOS khi sử dụng và nối ghép

1. Tất cả các đầu vào của cổng TTL để hở (không được nối) sẽ hoạt động như có logic 1 đặt tới đầu vào đó (do diốt bazơ – emitter của T_1 ứng với đầu vào này không được phân cực thuận). Trường hợp này gọi là thả nổi đầu vào. Khi không sử dụng 1 đầu vào nào đó của cổng TTL, ta có thể xử lý nó như trên hình 2.40.

- Với các cổng OR hay NOR chỉ dùng cách xử lý ở hình 2.40a hoặc nếu dùng cách xử lý như hình 2.40b phải nối đầu vào không sử dụng tới 0V để tạo logic 0 cho đầu vào này.



Hình 2.40

Cách xử lý đầu vào thứ 3 không sử dụng của cổng NAND 3 đầu vào.

– Khi có nhiều đầu vào đầu chung như hình 2.40a, đầu vào chung có được sẽ biểu thị một tải có dòng bằng tổng dòng các đầu vào được nối (trừ trường hợp cổng AND và NAND vì dòng chỉ do T_1 cung cấp, nếu đầu nối nhiều thì chỉ chia bớt dòng đi khi cổng kích thích ở logic 0).

Ví dụ : Có 4 cổng TTL nối như hình 2.41, các cổng thuộc họ 74LS có

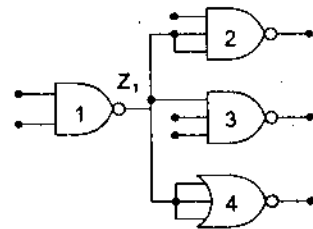
$$I_{IH} = 20\mu A ; I_{IL} = 0,4mA = 400\mu A.$$

Hãy xác định dòng tải mà đầu ra Z_1 đang kích thích.

Giải: Đầu ra cổng số 1 nối tới các đầu vào của 2 cổng NAND và 1 cổng NOR.

Như vậy khi Z_1 ở trạng thái cao ($Z_1 = 1$) mức tải của cổng 1 tại đầu ra của nó là 6 đầu vào của họ 74LS. Còn khi $Z_1 = 0$ mức tải chỉ còn 5 đầu vào (vì các cổng NAND ở chế độ thu dòng không phụ thuộc số lượng đầu vào của mỗi cổng) kết quả là: khi $Z_1 = 1$ cổng 2 nhận dòng là $2.20\mu A$, cổng 3 nhận dòng là $1.20\mu A$ cổng 4 nhận dòng là $3.20\mu A$ (hình 2.41).

Dòng kích thích của đầu ra cổng 1 tới các đầu vào của các cổng tải khi $Z_1 = 1$ là $40\mu A + 20\mu A + 60\mu A = 120\mu A$.

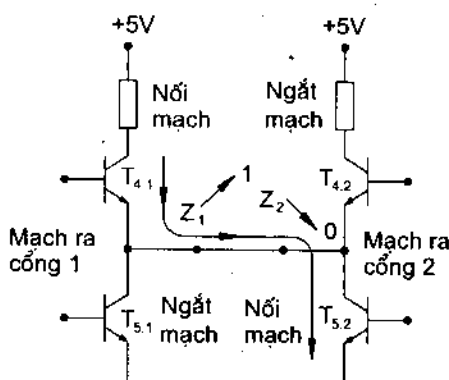


Hình 2.41. Cho ví dụ tính dòng tải của cửa ra cổng 1

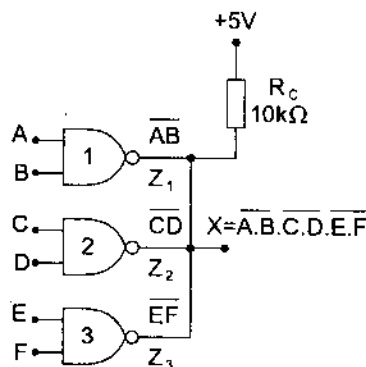
Khi $Z_1 = 0$ cổng 1 ở chế độ thu dòng của 5 cổng tải nên dòng thu tổng cộng là: $0,4mA + 0,4mA + 3.0,4mA = 2mA$.

2. Khi nối trực tiếp hai đầu ra của hai cổng TTL có cấu trúc mạch dạng hình 2.10 sẽ xảy ra xung đột. Ví dụ một cổng đầu ra Z_1 đang được kéo từ 0 lên 1 trong khi cổng kia đầu ra đang chuyển Z_2 từ 1 về 0 xuất hiện tình huống xung đột cao thấp và thông thường mức thấp thắng thế. Xem trên hình 2.42 ta thấy rõ điều này vì khi đó T_{S2} của cổng 2 nối mạch sẽ kéo mức ra của cổng 1 xuống mức thấp mặc dù T_{S1} của cổng 1 đang ngắt mạch và T_{41} đang nối mạch để "cố kéo" Z_1 lên 1.

Việc phân tích dẫn tới kết luận các cổng NAND có mạch điện dạng hình 2.10 với tầng ra như hình 2.42 không nên nối đầu ra của chúng trực tiếp với nhau.



Hình 2.42. Xung đột cao thấp xảy ra khi $Z_1 = 1$, $Z_2 = 0$: Z_2 thẳng



Hình 2.43. Nối các đầu ra của các cổng NAND TTL OC (hở cực góp) tạo ra hiệu ứng hàm AND

Nếu khi do nhu cầu sử dụng vẫn phải cần nối các đầu ra của cổng TTL một cách trực tiếp thì chọn mạch cấu trúc TTL có colectơ để hở rất thuận tiện cho việc ghép nối này. Hình 2.43 minh họa kiểu nối này.

Ta thấy rõ từ cấu trúc mạch điện cổng OC hình 2.32 đã xét, tranzitor T_5 tăng ra để hở colectơ “chờ” người sử dụng chọn điện trở colectơ mạch ngoài R_C nối vào, khi nối theo hình 2.43, 3 đầu ra dùng chung một điện trở tải colectơ tạo nên hiệu ứng hàm AND cho các trạng thái logic ở các đầu ra:

$$X = Z_1.Z_2.Z_3 = \overline{AB.CD.EF}$$

3. Kiểm soát đường đi của tín hiệu nhờ cổng loại đệm 3 trạng thái

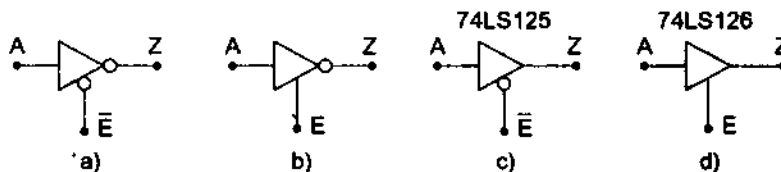
Trong mục 2.10.2 ta đã gặp cổng NAND loại có 3 trạng thái ở đầu ra : $Z = 1$ hay $Z = 0$ khi xuất hiện tín hiệu cho phép ở mức tích cực thấp $E = 0$ và trạng thái treo với trở kháng ra lớn (gọi trạng thái Hi-Z) khi tín hiệu cho phép ở mức $E = 1$.

Tương tự như vậy nếu chỉ có 1 đường tín hiệu vào A và:

– Đầu ra Z là tích cực thấp (đảo) ta có bộ đảo 3 trạng thái (ký hiệu quy ước trên hình 2.44a và b).

– Đầu ra Z là mức tích cực cao, ta có bộ đệm 3 trạng thái (ký hiệu quy ước cho trên hình 2.44c và d).

Các phân tử cổng loại này được sử dụng để kiểm soát (cho phép hay không cho phép) đường truyền của tín hiệu A nối hay ngắt mạch.



Hình 2.44. Bộ đảo 3 trạng thái (a,b) và bộ đệm 3 trạng thái (c,d)

- + Với cổng (2.44a) : $Z = \bar{A}$ khi $\bar{E} = 0$, Z ở Hi-Z khi $\bar{E} = 1$
- + Với cổng (2.44b) : $Z = \bar{A}$ khi $E = 1$ và Z ở Hi-Z khi $E = 0$
- + Với cổng (2.44c) : $Z = A$ khi $\bar{E} = 0$ và Z ở Hi-Z khi $\bar{E} = 1$
- + Với cổng (2.44d) : $Z = A$ khi $E = 1$ và Z ở Hi-Z khi $E = 0$

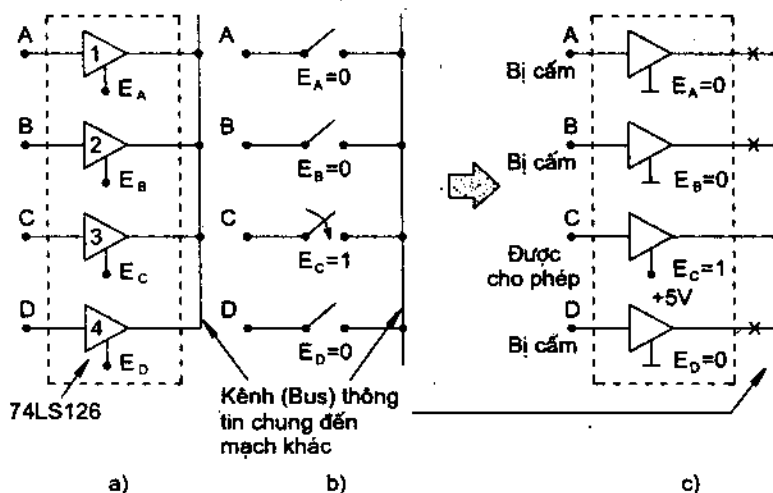
Ta có bảng trạng thái với cổng (2.44c và d) như sau:

Bảng trạng thái của
bộ đệm 3 trạng thái
dùng cổng 74LS125
(Mỗi IC chứa 4
cổng độc lập)

$\bar{E}$	Z
0	A
1	Hi-Z

Bảng trạng thái của
bộ đệm 3 trạng thái
dùng cổng 74LS126
(mỗi IC chứa 4
cổng độc lập)

E	Z
0	Hi-Z
1	A



Hình 2.45. IC đệm 74LS126 chọn dữ liệu 4 đường A,B,C,D đặt tới kênh chung thông qua tín hiệu điều khiển chọn E_A, E_B, E_C, E_D . Khi $E_C = 1$ và $E_A = E_B = E_D = 0$ chỉ có C xuất hiện ở cổng ra nối với kênh chung

Tên hình 2.45 là một ví dụ dùng 4 bộ đệm của IC 74LS126 kiểm soát 4 đường tín hiệu A, B, C và D dẫn tới một kênh thông tin (Bus) chung với điều kiện mỗi lần xuất hiện $E_x = 1$ ($X = A, B, C, D$) chỉ cho phép một trong các đầu ra của các cổng 1, 2, 3, 4 được nối tới kênh chung và như vậy chỉ 1 luồng thông tin A (hoặc B, hoặc C, hoặc D) được truyền tới kênh chung này.

4. Các đặc điểm quan trọng của IC cổng loại CMOS khi sử dụng

- Tính tương thích chân (bố trí cấu hình chân giống nhau cho các seri), tính tương thích về chức năng (nhóm IC cùng thực hiện một chức năng logic) và tính tương thích cao về điện (có thể nối ghép trực tiếp với nhau).

- Có các mức dòng điện và điện thế giới hạn tại đầu vào, đầu ra ở hai trạng thái mức cao và mức thấp tối ưu, giới hạn chống nhiễu do đó lớn.

- Công suất tiêu thụ 1 chiều đặc biệt nhỏ (10^{-9} W đối với mỗi cổng), tuy nhiên sẽ tăng mạnh theo tần số chuyển mạch do có mặt điện dung tải và điện dung ra của MOS (C_{DS} , C_{DG} , C_{GS} của cổng sau...) đặc biệt lúc chuyển từ mức thấp lên mức cao (dòng nạp cho tụ tính trung bình tăng nhanh theo tần số đóng ngắt mạch). Đây là nhược điểm quan trọng của các cổng MOS mà ở tần số làm việc cỡ 10^6 Hz phải chú ý đặc biệt.

- Hệ số tải: do có mặt điện dung vào của cổng tải (cỡ từ 0,5pF đến 5pF, $1\text{pF} = 10^{-12}\text{F}$ đối với mỗi đầu vào) sẽ giới hạn số lượng đầu vào (mà không phải do dòng tiêu thụ của mỗi đầu vào tải vì là vô cùng bé). Đầu ra của cổng kích thích phải thực hiện nạp, phóng cho tất cả các điện dung tải nối song song với nhau do đó thời gian chuyển mạch của đầu ra tăng tỷ lệ với số đầu vào tải nối với nó, thường thời gian trễ là $3\text{ns} = 3 \cdot 10^{-9}\text{s}$ với mỗi đầu vào tải. Như vậy một cổng MOS khi không tải có thời gian trễ là 20ns thì khi có 30 tải mắc tới nó thời gian trễ sẽ là $(20\text{ns} + 30 \cdot 3\text{ns}) = 110\text{ns}$ với điện dung tải tổng cộng là $30 \cdot 0,5\text{pF} = 15\text{pF}$.

- Tốc độ chuyển mạch nhanh do điện trở đầu ra thường nhỏ ($< 1\text{k}\Omega$).

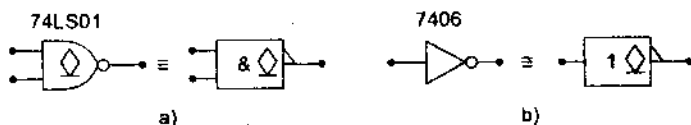
- Không được phép thả nổi các đầu vào không sử dụng đến, phải nối chúng tới 0V hay $+V_{DD}$ hay với đầu vào khác tùy yêu cầu chức năng logic của cổng.

- Điện trở đầu vào cao gây tích tụ hạt tĩnh điện phát sinh điện thế có thể đủ lớn đánh thủng lớp điện môi mỏng giữa G và kênh dẫn. Người ta bảo vệ bằng lưới diốt điện trở ở đầu vào (chế tạo sẵn).

- Trạng thái tranzitor lưỡng cực ký sinh (npn hay pnp hình thành tự nhiên do đặc điểm công nghệ chế tạo MOS) bị nối mạch vĩnh viễn từ một lúc nào đó gây ra 1 dòng lớn phá hủy IC (trạng thái Latch-Up) cần phải được phòng ngừa với mọi loại cổng MOS.

2.11.4. Các bộ đệm, bộ điều khiển hở cực colecto

Các IC đệm, điều khiển cho phép mức dòng ra và (hoặc) điện thế ra lớn hơn các cổng logic bình thường, dùng để kích thích tải sử dụng dòng hay điện thế lớn hơn.

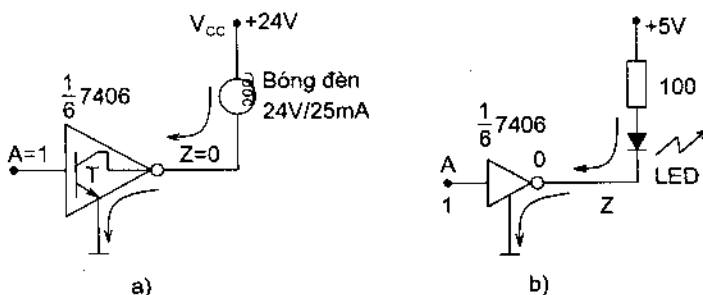


Hình 2.46. Ký hiệu quy ước chuẩn và ký hiệu chữ nhật của
a) Cổng NAND ; b) Cổng đảo có collector để hở

Ví dụ IC 7406 (7407) gồm 6 bộ đảo (bộ đệm) có khả năng thu nhận dòng tới đa 40mA khi đầu ra ở mức thấp và tranzito đầu ra có thể nối tới mức điện áp trên +5V (tới +30V) qua điện trở tải như 1 bóng đèn sợi nung 24V/ 25mA. Ký hiệu quy ước của các phần tử có collector để hở này theo chuẩn IEEE/ ANSI có ký pháp "◻" bên trong như thể hiện ở hình 2.46.

Khi nối với tải cao áp hay cao dòng các cổng này được nối trực tiếp vào collector của tranzito đầu ra như thể hiện trên hình 2.47.

Khi $A = 1$ (mức tích cực cao) $Z = 0$ (mức tích cực thấp) xuất hiện dòng từ $+V_{CC}$ qua tải, qua tranzito mạch ra có cực góp (collector) đã nối với tải về mạch đất 0V.



Hình 2.47. Nối trực tiếp cổng đệm điều khiển tới tải
a) Điều khiển tải cao áp, cao dòng ; b) Điều khiển tải LED bình thường (10-20mA)

2.11.5. Nối ghép cổng TTL với CMOS và CMOS với TTL

I. Khi cần nối các đầu ra của một mạch logic (hay một hệ thống) đến các đầu vào của mạch hay hệ thống khác có đặc tính về điện khác nhau ta cần thực hiện giao diện giữa hai mạch hay hai hệ thống và nhờ các cổng giao diện tương ứng. Vậy nhiệm vụ của mạch (cổng) giao diện là kết nối giữa mạch kích thích (phía đầu ra) nơi cung cấp tín hiệu với phía đầu vào mạch tải nơi nhận tín hiệu đến. Khi nối các cổng cùng họ công nghệ thường việc ghép nối đã được thiết kế tính toán sẵn bởi nhà sản xuất nên vấn đề còn lại chỉ là giới hạn hệ số tải của mỗi đầu ra kích thích. Khi nối các cổng khác họ, ví dụ TTL với CMOS hoặc ngược lại, cần chú ý các thông số dòng điện và điện thế tiêu chuẩn giữa 2 họ để chọn cách thức giao diện cho phù hợp.

2. Trường hợp TTL kích thích CMOS là tải. Để tiện lợi cho việc phân tích ta có bảng (2.3) và (2.4) so sánh 4 tham số dòng điện và 6 tham số điện thế quan trọng nhất của hai họ phổ biến và quan trọng nhất (CMOS và TTL).

Bảng 2.3. CÁC DÒNG ĐIỆN VÀO (mức thấp, mức cao) VÀ DÒNG ĐIỆN RA (mức thấp, mức cao) CHO MỘT SỐ LINH KIỆN CHUẨN CỦA 2 Họ CMOS VÀ TTL

Thông số	CMOS				TTL				
	4000B	74HC	74AC (L)	74AHC	74	74LS	74AS	74ALS	74F
I_{IHmax}	1 μ A	1 μ A	1 μ A	1 μ A	40 μ A	20 μ A	200 μ A	20 μ A	20 μ A
I_{ILmax}	1 μ A	1 μ A	1 μ A	1 μ A	1,6mA	0,4mA	2mA	100mA	0,6mA
I_{OHmax}	0,4mA	4mA	24mA	8mA	0,4mA	0,4mA	2mA	400mA	1,0mA
I_{OLmax}	0,4mA	4mA	24mA	8mA	16mA	8mA	20mA	8mA	20mA

Chú ý

* 1 μ A = 10^{-6} A ; 1mA = 10^{-3} A; V_{cc} = 5V ;

I_{IHmax} dòng vào cực đại ở mức cao ; I_{ILmax} dòng vào cực đại ở mức thấp ;

I_{OHmax} dòng ra cực đại ở mức cao ; I_{OLmax} dòng ra cực đại ở mức thấp

* Với TTL

LS: tốc độ cao dùng diốt Shotky ; AS tốc độ siêu cao dùng diốt Shotky

ALS tốc độ cao, công suất thấp ; F tốc độ siêu cao, thời gian trễ cực nhỏ

* Với CMOS

Chữ HC biểu thị CMOS tốc độ cao (cỡ 10 lần seri LS)

Chữ ACL biểu thị khả năng chống nhiễu tốt, tốc độ nhịp cao, trễ nhỏ

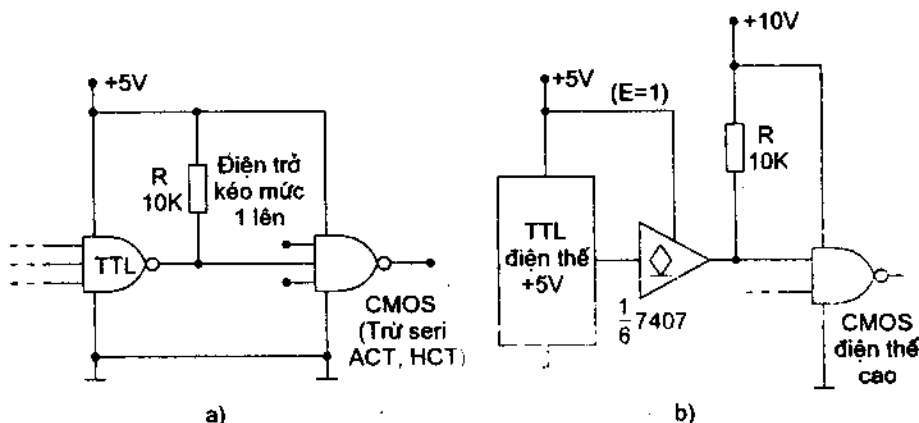
Chữ AHC biểu thị tốc độ cực cao (3 lần HC) công suất, dòng tiêu thụ thấp.

Bảng 2.4. CÁC MỨC ĐIỆN THẾ ĐẦU VÀO, ĐẦU RA (V) KHI $+V_{cc}$ = 5V = $+V_{DD}$ CỦA 2 Họ CMOS VÀ TTL

Thông số	CMOS							TTL			
	4000B	74HC	74HCT	74AC	74ACT	74AHC	74AHCT	74	74LS	74AS	74ALS
$V_{IH}(min)$	3,5	3,5	2,0	3,5	2,0	3,85	2,0	2,0	2,0	2,0	2,0
$V_{IL}(max)$	1,5	1,0	0,8	1,5	0,8	1,65	0,8	0,8	0,8	0,8	0,8
$V_{OH}(min)$	4,95	4,9	4,9	4,9	4,9	4,4	3,15	2,4	2,7	2,7	2,7
$V_{OL}(max)$	0,05	0,1	0,1	0,1	0,1	0,44	0,1	0,4	0,5	0,5	0,4
V_{NH}	1,45	1,4	2,9	1,4	2,9	0,55	1,15	0,4	0,7	0,7	0,7
V_{NL}	1,45	0,9	0,7	1,4	0,7	1,21	0,7	0,4	0,3	0,3	0,3

Vấn đề cần giải quyết là cổng kích thích có thỏa mãn yêu cầu về dòng điện và điện thế của các cổng tải hay không? Theo bảng (2.3) dòng điện khi dùng TTL kích thích CMOS là luôn thỏa mãn, tuy nhiên theo bảng (2.4) các yêu cầu điện thế đầu

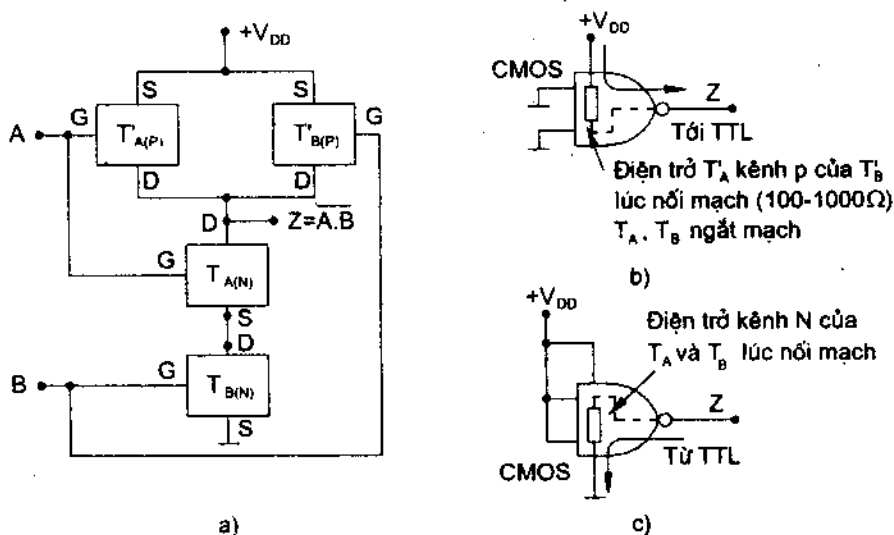
ra TTL và đầu vào CMOS khác nhau (ví dụ, V_{OHmin} của TTL quá thấp so với V_{IHmin} của CMOS). Giải pháp chung là nâng điện thế ra của cổng TTL kích thích lên đến mức đầu vào CMOS chấp nhận được thể hiện cách nối (giao diện) trên hình (2.48a và b).



Hình 2.48. Các giải pháp dùng mạch giao diện kéo mức 1 của TTL lên cho phù hợp với yêu cầu điện thế vào của CMOS

- a) Dùng điện trở kéo mức cao lên khi TTL và CMOS cùng điện thế nuôi
- b) Dùng bộ đệm hở cực collectơ thực hiện khi CMOS có điện thế nuôi cao hơn.

3. Trường hợp CMOS kích thích, TTL là tải



Hình 2.49. Các chế độ phun dòng (b) và hút dòng (c) của đầu ra của cổng NAND CMOS (a) đối với tải TTL

– Khi ở trạng thái mức cao, đầu ra của CMOS phun dòng ra tải hình 2.49b. Còn khi ở mức thấp đầu ra của CMOS hút dòng từ cổng tải, hình 2.44a biểu diễn cấu trúc tầng ra của cổng NAND CMOS.

Ta thấy rõ từ hình (2.49b và c) hai chế độ khi $Z = 1$ điện trở nối mạch của T' kênh p khép dòng phun từ $+V_{DD}$ qua R_{DS} nối tới đầu ra cổng kích thích và hướng tới các đầu vào của cổng tải TTL. Còn khi $Z = 0$ dòng hút từ tải TTL sẽ tới đầu ra cổng CMOS qua T_A và T_B lúc đó nối mạch có điện trở thấp (vài trăm Ω) tới đất 0V.

Kết quả việc phân tích trên là:

– Khi đầu vào TTL ở mức cao, ở trạng thái cao V_{OHmin} của CMOS hoàn toàn vượt giá trị V_{IHmin} của TTL đủ đáp ứng yêu cầu của tải (bảng 2.4). Theo bảng 2.3 đầu ra CMOS ở trạng thái này cũng hoàn toàn đủ cung cấp dòng cho tải $I_{OH} > I_{IH}$.

– Khi TTL ở mức thấp có dòng vào tương đối cao (bảng 2.3) Seri IC 74HC, 74HCT có dòng thu nhận ở đầu ra tương đối cao (tối đa 4mA) nên chúng dễ dàng kích thích 1 tải TTL đơn lẻ (trừ loại IC4000B có dòng $I_{OLmax} = 0,4mA$ tương đối nhỏ nên sẽ gặp hạn chế).

Ví dụ : Đầu ra 74HC có thể kích thích được bao nhiêu đầu vào TTL 74LS ? bao nhiêu đầu vào IC4000B ? bao nhiêu đầu vào 74ALS, 74AS ? Để trả lời câu hỏi trên ta tra bảng 2.3 được: loại 74LS có $I_{ILmax} = 0,4mA$.

loại 74HC có $I_{OLmax} = 4mA$ (thu nhận tối đa)

– Vậy đầu ra của IC CMOS 74HC có thể kích thích được

$$\frac{I_{OLmax}}{I_{ILmax}} = \frac{4mA}{0,4mA} = 10 \text{ tải IC TTL 74LS}$$

– Tuy nhiên với IC 4000B $I_{OL} = 0,4mA$ nên chỉ đủ kích thích một tải TTL loại 74LS.

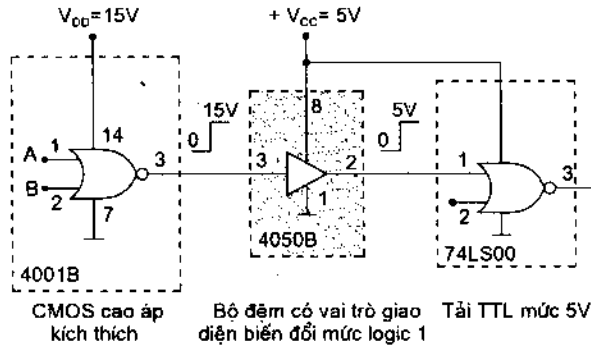
– Với tải là loại IC 74ALS có $I_{ILmax} = 100 \mu A$, do vậy IC 74HC có thể kích thích

$$\frac{4mA}{0,1mA} = 40 \text{ đầu vào cổng loại 74 ALS.}$$

– Khi tải là IC 74 AS có $I_{ILmax} = 2mA$, số cổng 74 AS có thể mắc phía sau là :

$$\frac{I_{OL(74HC)}}{I_{IL(74AS)}} = \frac{4mA}{2mA} = 2$$

4. CMOS điện thế cao kích thích tải TTL. Đa số các đầu vào TTL không chịu được điện thế trên 7V (trừ seri 74LS có loại chịu được 15V) trong khi mức cao của CMOS nhiều loại là +15V khi đó cần mạch giao diện biến đổi mức cao đầu ra của CMOS thành mức 5V nối tới đầu vào TTL. Mạch điện hình 2.50 giải quyết nhiệm vụ trên nhờ dùng cổng đệm IC4050B.



Hình 2.50. Biến đổi mức logic 1 cao áp của CMOS thành mức 5V của TTL tải

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 2

2.1. Hãy nêu các tính chất đặc trưng của một cổng logic và giải thích rõ ý nghĩa của tham số đã nêu. Mục đích việc đưa ra các tham số này?

2.2. a) Với một cổng AND 5 đầu vào A, B, C, D, E tìm giá trị các biến vào để lỗi ra Z của nó nhận được ở mức cao? Hãy viết biểu thức của hàm Z.

b) Nếu cho rằng : Với các đầu vào có giá trị giống nhau, đầu ra của một cổng AND và một cổng OR luôn luôn có giá trị khác nhau có đúng không? giải thích?

c) Xây dựng một mạch điện cổng AND 3 đầu vào dùng điốt tương tự như hình 2.4

2.3. Cho mạch điện cổng hình B2.3

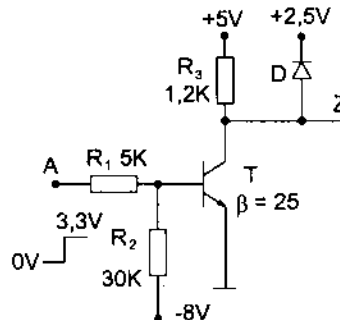
a) Giải thích hoạt động và chức năng logic của mạch

b) Với mức vào thấp là 0V cao là 3,3V

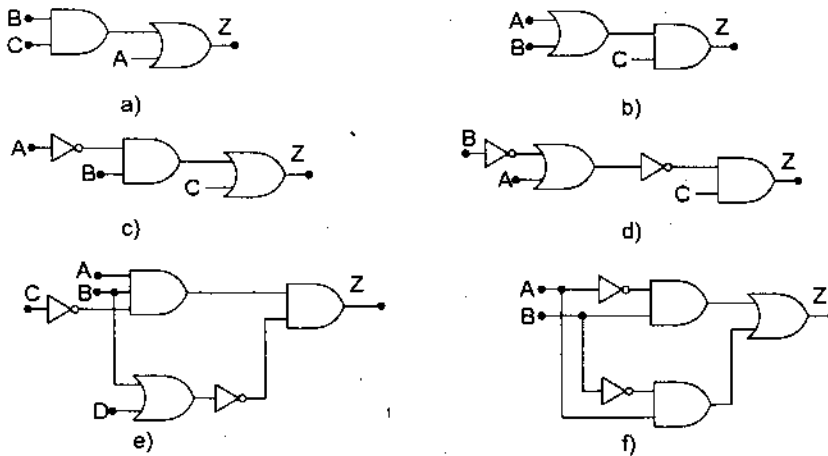
Hãy xác định các mức ra tương ứng.

Khi nào T ở chế độ bão hoà? Hãy kiểm tra điều kiện bão hoà của T.

2.4. Hãy xác định biểu thức hàm ra Z ở các cấu trúc (hình B2.4):



Hình B2.3



Hình B2.4

2.5. Cho hàm logic $Z = AC + \overline{BC} + \overline{ABC} + \overline{BA}$

a) Hãy xây dựng cấu trúc sơ đồ logic thực hiện Z từ các cổng OR, AND và NOT.

b) Hãy đưa Z về dạng tổng các tích các biến $Z = \sum m_i$ và lập bảng chân lý cho Z.

2.6. a) Bằng cấu trúc các ký hiệu hàm NAND, hãy chứng minh có thể tạo ra các hàm logic cơ sở còn lại: AND, OR, NOT và NOR.

b) Câu hỏi tương tự cho hàm NOR, hãy tạo ra các hàm còn lại chỉ từ NOR.

2.7. Hãy viết bảng chân lý, biểu thức hàm logic cho một cổng NOR có 3 đầu vào và cho một cổng NAND có 3 đầu vào.

Mạch điện logic thực hiện chúng có gì khác so với các mạch điện có 2 đầu vào đã xét ở các hình 2.8a và c, hình 2.10a và b.

2.8. Cho hàm 3 biến $Z = AB + BC + CA$

a) Hãy viết lại biểu thức của Z theo cấu trúc hàm NAND và theo cấu trúc hàm NOR.

b) Xây dựng sơ đồ logic thực hiện Z chỉ bởi các cổng NAND và chỉ bởi các cổng NOR.

c) Nếu cần xây dựng cấu trúc hàm $(\overline{A+B})(\overline{B+C})(\overline{C+A}) = Z_1$ có những cách nào? Có thể dựa vào kết quả của câu b được không? vì sao?

2.9. a) Tại sao phải sử dụng các ký hiệu logic thay thế tương đương? Ý nghĩa của khái niệm mức logic tích cực là gì?

b) Hãy giải thích hoạt động theo ý nghĩa ký hiệu chuẩn của cổng NOR và ý nghĩa này thay đổi như thế nào nếu dùng ký hiệu tương đương thay thế của cổng NOR.

c) Câu hỏi tương tự câu b) cho hai ký hiệu chuẩn và tương đương thay thế của cổng AND.

2.10. Hãy vẽ các sơ đồ gồm chỉ các phần tử tương đương thay thế của các mạch logic bài tập 2.4 và giải thích ý nghĩa các đường tín hiệu vào và đường tín hiệu ra, đường tín hiệu trung gian của các cấu trúc mới thành lập.

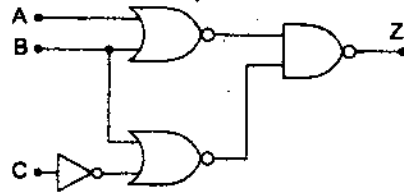
2.11. Cho mạch hình B2.11

a) Viết biểu thức logic đầu ra cho Z.

b) Xây dựng bảng chân lý của Z.

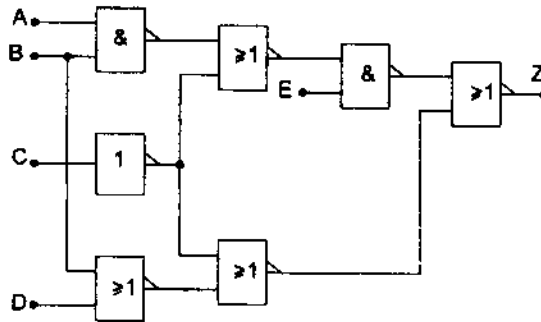
c) Chuyển đổi ký hiệu thay thế ở những chỗ cần thiết để có mức logic tích cực phù hợp với yêu cầu mức tích cực ở đầu ra là thấp.

d) Tương tự khi yêu cầu mức ra là tích cực cao.



Hình B2.11

2.12. Xác định biểu thức logic cho hàm Z với cấu trúc hình B2.12.



Hình B2.12

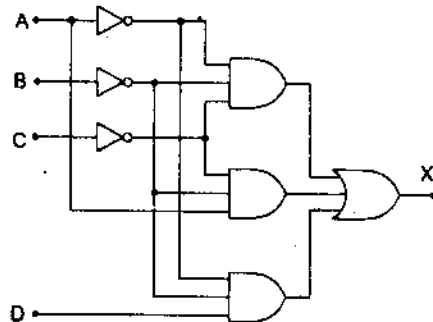
2.13. Giả thiết yêu cầu đầu ra của mạch hình B2.12 tích cực ở mức thấp. Hãy vẽ lại mạch một cách thích hợp để mô tả rõ hơn hoạt động của nó. (Yêu cầu các đường ghép tín hiệu giữa các cổng phải phù hợp mức tích cực với nhau).

2.14. Cho mạch hình B2.14

a) Viết biểu thức của X.

b) Thiết lập bảng chân lý của X theo các biến A, B, C, D.

c) Vẽ lại cấu trúc của mạch khi chỉ sử dụng cổng NAND thực hiện X.



Hình B2.14

2.15. Cổng NAND hở collector là gì? Tại sao phải để hở collector?

Hãy giải thích tình huống xung đột cao, thấp khi nối hai đầu ra của hai cổng NAND bình thường? Tại sao mức 0 đầu ra của bất kỳ cổng nào trong hai cổng trên sẽ thắng thế?

2.16. Cổng NAND có đầu ra 3 trạng thái có đặc điểm gì ? ý nghĩa của đầu vào cấm? Tại sao phải có trạng thái thứ 3 : Hi. Z?

Nêu ưu điểm quan trọng nhất của cổng logic loại 3 trạng thái trong việc điều khiển đường truyền tín hiệu.

2.17. Bộ đệm và bộ đảo 3 trạng thái có gì khác NAND 3 trạng thái?

Hãy vẽ ký hiệu quy ước bộ đệm và bộ đảo 3 trạng thái dùng mức tích cực cao hoặc mức tích cực thấp điều khiển trạng thái của nó.

Hãy cho một ví dụ việc sử dụng cổng 3 trạng thái để kiểm soát các luồng số liệu A, B, C, D trạng thái tích cực thấp ở đầu ra của chúng.

2.18. Cổng XOR (gọi là hoặc tuyệt đối) thực hiện hàm logic gì ? ký hiệu cổng? Bảng trạng thái của hàm logic mà cổng thực hiện?

Hãy chứng minh các tính chất sau của hàm hoặc tuyệt đối:

- a) $A \oplus B = B \oplus A$ (luật hoán vị)
- b) $(A \oplus B) \oplus C = A \oplus (B \oplus C)$ (luật kết hợp)
- c) $A (B \oplus C) = AB \oplus AC$ (luật phân bố)
- d) Nếu $A \oplus B = C$ Thì $A \oplus C = B$ và $B \oplus C = A$ (luật đổi chỗ nhân quả)

2.19. a) Hãy viết biểu thức của hàm $\overline{F_{XNOR}} = \overline{A \oplus B}$ gọi tên là hàm tương đương (XNOR) có ký hiệu quy ước cho ở hình B2.19 (xem hình 2.34c hoặc xem bảng 1.4).

b) Viết bảng chân lý và nhận xét kết quả thu được của hàm tương đương.

c) Hãy chứng minh rằng:

$$Z = \overline{(A \oplus B)(C \oplus D)} = \overline{A}.\overline{B} + AB + \overline{C}.\overline{D} + CD$$



Hình B2.19

d) Vẽ sơ đồ logic của hàm $X = A \oplus B \oplus C \oplus D$.

2.20. a) Hãy trình bày hai phương thức hoạt động khi thực hiện ghép nối các cổng logic với nhau (cổng phía trước gọi là cổng kích thích, cổng phía sau nối với đầu ra của cổng trước gọi là cổng tải).

b) Phương pháp xác định hệ số tải ? cách thực hiện như thế nào ?

2.21. Các đặc điểm quan trọng khi sử dụng các đầu vào của cổng TTL:

a) Hãy nêu cách xử lý đầu vào thừa không dùng đến đối với cổng TTL NAND

b) Xử lý các đầu vào không dùng đến của cổng OR, NOR có gì khác cổng NAND TTL?

c) Các đầu ra của cổng TTL OC (hở colectơ) được nối với nhau như thế nào? Hãy chứng minh hiệu ứng hàm AND khi thực hiện việc nối trên qua ví dụ ghép 2 đầu ra của 2 cổng NAND OC có các biến vào A, B và C, D tác động.

2.22. a) Nêu các đặc điểm quan trọng của IC CMOS ? Cần quan tâm tới các tham số gì của chúng ?

b) Khi sử dụng IC TTL kích thích các tải là IC CMOS cần quan tâm tới các tham số gì của chúng ? Hãy nêu lý do tại sao phải kéo giá trị mức cao của cổng kích thích lên trước khi đưa tới tải?

Nêu các biện pháp thực hiện trong hai trường hợp:

- Các cổng có chung giá trị nguồn 1 chiều.
- Cổng CMOS có nguồn nuôi 1 chiều cao hơn cổng TTL.

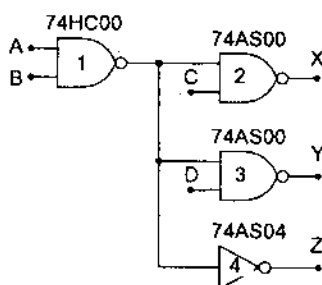
2.23. a) Khi sử dụng IC cổng CMOS kích thích tải là IC TTL cần quan tâm tới những tham số gì của đầu ra kích thích và những tham số gì của các đầu vào tải?

b) Khi nào cần dùng giao diện là IC cổng đệm? Cho ví dụ minh họa.

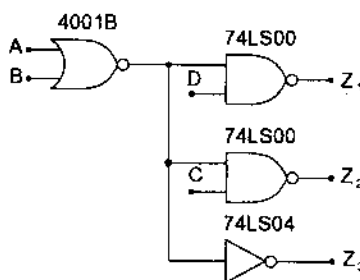
2.24. Cho mạch điện hình B2.24

Hãy sử dụng các bảng (2.3) và (2.4) tìm xem có điều gì không đúng trong cấu trúc đã cho ? phải sửa lại thế nào để đảm bảo vẫn nhận được các hàm ra X, Y, Z như cũ?

2.25. Tìm lỗi trong cấu trúc mạch hình B2.25? Cần sửa lại như thế nào để đảm bảo các hàm ra Z_1 , Z_2 , Z_3 như ban đầu?



Hình B2.24

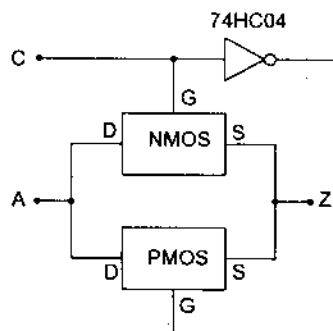


Hình B2.25

2.26. a) Giải thích hoạt động của mạch CMOS hình B2.26 dùng 1/4 IC 4016/ 74HC4016 khi tín hiệu điều khiển C có 2 mức 0V và $+V_{DD}$ tác động vào mạch đã cho?

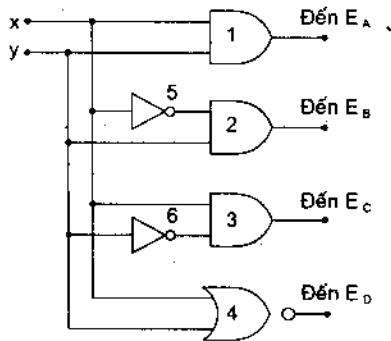
b) Khi có 4 tín hiệu A_1 , A_2 , A_3 , A_4 đầu vào 4 mạch điện hình B2.26 đã cho dưới sự điều khiển của các tín hiệu C_1 , C_2 , C_3 và C_4 tương ứng, các cổng ra Z đầu chung với nhau. Hãy tìm điều kiện để tại lối ra nhận được chỉ 1 trong các tín hiệu A_i ở lối vào.

2.27. Người ta sử dụng mạch điện hình B2.27 để cung cấp đầu vào cho phép E_A hoặc E_B hoặc E_C hoặc E_D cho mạch điện hình 2.45c hoạt động.

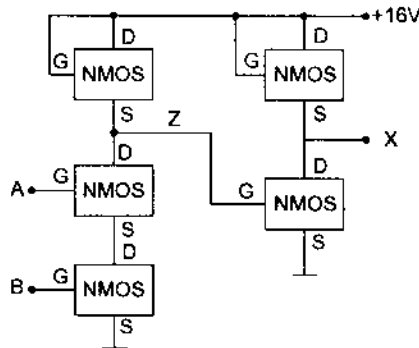


Hình B2.26

Hãy thiết lập bảng trạng thái của mạch điện trên để chứng minh rằng mạch B2.27 đáp ứng được yêu cầu hoạt động của mạch hình 2.45c.



Hình B2.27

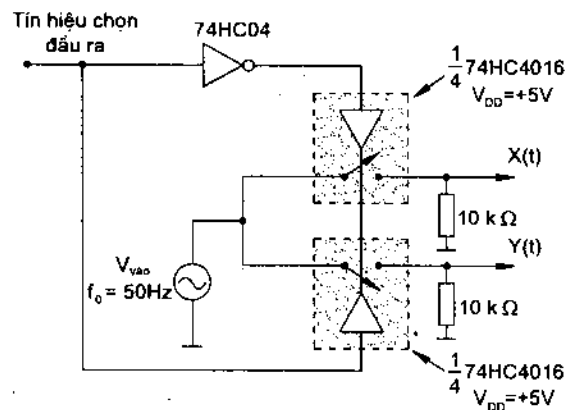


Hình B2.28

2.28. Cho mạch điện hình B2.28 dùng NMOS.

- Xác định giá trị hàm ra X theo các biến vào A, B (mức cao là +16V).
- Lập bảng trạng thái của hàm X.
- Thay thế mạch điện trên bằng sơ đồ các ký hiệu cổng logic chuẩn.
- Phân tích dòng điện khi $Z = 1$ và khi $Z = 0$ (chế độ phun và chế độ hút dòng).

2.29* Để lồng (gửi) một bit nhị phân giá trị 0 hay 1 vào một sóng hình sin tần số f_0 , người ta sử dụng mạch điện hình B2.29 gọi là một mạch điều biến số ASK (điều này thường gặp trong kỹ thuật thông tin số trong đó tín tức thể hiện nội dung qua các ký tự

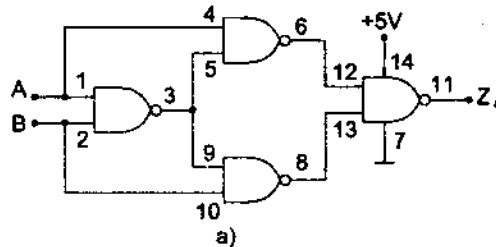


Hình B2.29

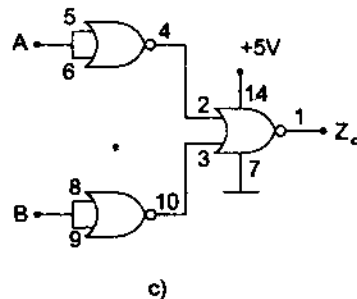
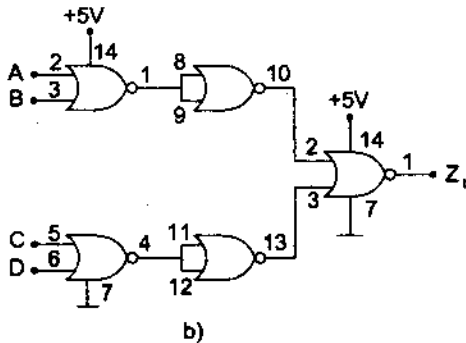
bất 0,1 còn sóng hình sin được gọi là sóng mang hay sóng tải tin). Hãy giải thích nguyên tắc hoạt động của mạch hình B2.29 với các cổng chuyển mạch 74HC4016.

Hãy vẽ dạng sóng $X(t)$ và $Y(t)$ tại các đầu ra tương ứng khi xuất hiện tín hiệu chọn là 1 dây xung nhị phân 1001 có chu kỳ 1 bit là $0,1s = 100ms$. Tín hiệu hình sin có tần số là 50 Hz biên độ $V_m = 3V$ (sắp xếp đồ thị theo trục thời gian thẳng hàng).

2.30. a) Cho mạch điện hình B2.30a. Hãy viết biểu thức hàm ra Z_a lập bảng trạng thái theo các giá trị đầu vào A và B có thể có.

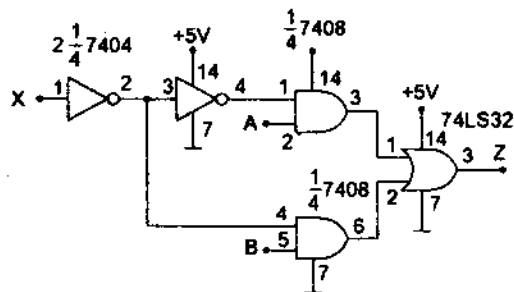


b) Câu hỏi tương tự cho mạch hình B2.30b dùng 2 cổng IC7402 và hình B2.30c dùng 1 cổng IC7402.



Hình B2.30

2.31. a) Cho mạch điện hình B2.31. Các luồng dữ liệu A hoặc B tùy theo tín hiệu chọn X ở mức cao hay thấp sẽ được đặt tới cổng ra Z. Hãy giải thích hoạt động và thiết lập bảng chân lý trạng thái của Z (theo 3 tín hiệu vào X, A, B).



Hình B2.31

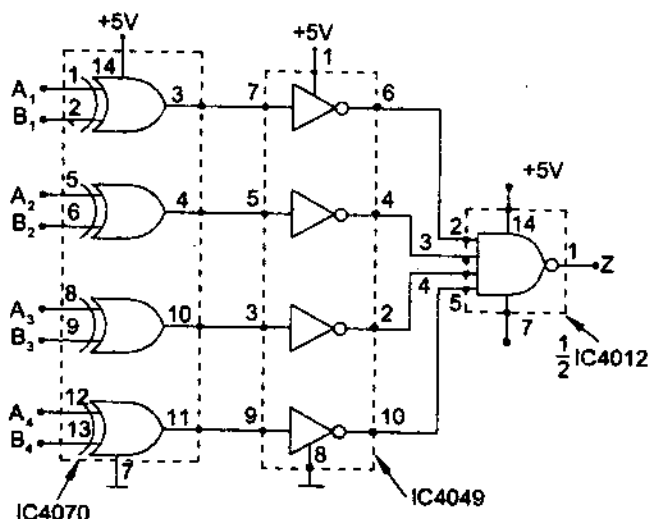
b) Hãy chuyển sơ đồ về dạng ký hiệu tương đương thay thế (phù hợp trạng thái tích cực cao của Z theo yêu cầu).

2.32* Để so sánh 2 bit nhị phân A và B, cần 3 đầu ra thể hiện các trạng thái kết quả $Z_1 = 1$ khi $A = B$, $Z_2 = 1$ khi $A > B$ và $Z_3 = 1$ khi $A < B$.

a) Hãy thiết lập bảng trạng thái gồm có 2 đầu vào A, B và 3 đầu ra Z_1, Z_2, Z_3 của cổng so sánh loại này.

b) Giải thích hoạt động của mạch điện hình B2.32 tìm quan hệ logic của hàm ra Z với các biến vào là

2 số 4 bit nhị phân 8421 : $A_4A_3A_2A_1$ và $B_4B_3B_2B_1$.

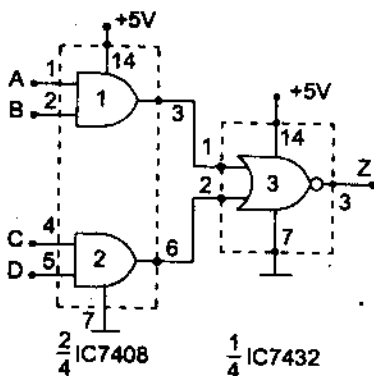


Hình B2.32

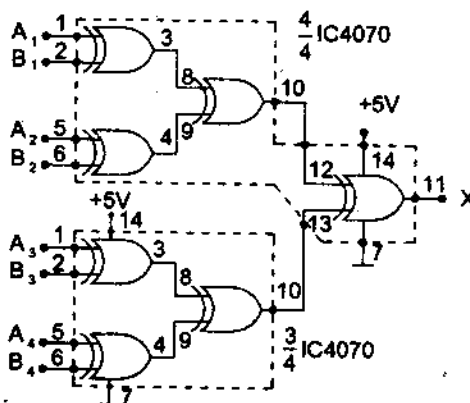
2.33. Cho mạch điện cổng hình B2.33 gọi tên là cổng AOI

Hãy thiết lập bảng chân lý của Z theo A, B, C, D và từ đó giải nghĩa tên cổng đã cho. (Các cổng 1 và 2 dùng 2.1/4 IC7408. Cổng 3 dùng 1/4 IC 7432)

2.34. Hãy thiết lập hàm ra X của cấu trúc hình B2.34 sử dụng hai IC4070, viết bảng trạng thái của X theo các biến vào.



Hình B2.33



Hình B2.34

Chương 3

CÁC MẠCH LOGIC TỔ HỢP (MSI)

Trong phần này chúng ta sẽ xét tới các cổng logic tổ hợp được tích hợp trong một chip IC cỡ vừa (MSI) có chứa khoảng vài chục tới hàng trăm các cổng logic cơ sở đã được xét đến ở chương 2. Nhóm linh kiện này được chế tạo nhằm thực hiện một số các hoạt động thu nhận, truyền tải, biến đổi các dữ liệu thông tin nhị phân, xử lý chúng theo một cách thức nào đó, bao gồm các nhiệm vụ chủ yếu sau:

- Mã hoá và giải mã các luồng ký tự nhị phân (gọi chung là các quá trình biến đổi mã) mà phần lý luận đã đề cập tới ở chương 1.
- Dồn kênh và phân kênh để chọn hoặc chia tách các luồng số nhị phân theo những yêu cầu nhất định để định tuyến cho chúng trong việc truyền dẫn thông tin.
- Các phép so sánh số để đánh giá định tính và định lượng trọng số của các số nhị phân.
- Truyền dữ liệu qua kênh thông tin chung (truyền tới bus dữ liệu).
- Logic tổ hợp có thể lập trình được PLA.
- Quan sát và phân tích để xử lý lỗi trong các mạch số...

3.1. ĐẶC ĐIỂM CƠ BẢN CỦA MẠCH LOGIC TỔ HỢP

3.1.1. Phân loại. Căn cứ vào đặc điểm và chức năng logic, các mạch số được chia thành 2 loại chính: mạch tổ hợp và mạch dãy (tuần tự).

Khi trị số ổn định của tín hiệu đầu ra của một mạch logic ở một thời điểm bất kỳ chỉ phụ thuộc vào tổ hợp các giá trị của tín hiệu số đầu vào ở thời điểm đó ta gọi mạch đang xét là mạch logic tổ hợp. Các trạng thái của mạch điện ở trước thời điểm xét (trước khi có tác động ở đầu vào) không ảnh hưởng gì tới trạng thái đầu ra. Đặc điểm quan trọng là mạch tổ hợp được cấu trúc chỉ từ các cổng logic cơ bản đã xét.

3.1.2. Nhiệm vụ

Hai nhiệm vụ thường gặp với người quan tâm nghiên cứu các mạch tổ hợp là:

1. Phân tích chức năng logic của một mạch đang xét đã có sẵn gồm các bước chính sau:

- Tuần tự từ các đầu vào đến các đầu ra, viết biểu thức hàm.
- Rút gọn hàm thu được bằng phương pháp đại số hay phương pháp hình vẽ (đồ thị các nơ).
- Thiết lập bảng chân lý biểu thị tất cả các trạng thái đầu vào có thể có và trạng thái đầu ra tương ứng.

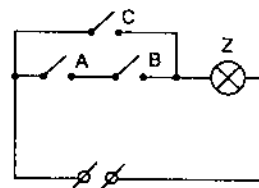
2. Thiết kế mạch logic tổ hợp

Xuất phát từ bài toán nhiệm vụ logic muốn đạt được, từng bước tìm ra mạch điện tử số thực hiện.

Quá trình thiết kế bao gồm 4 bước chính sau:

- Phân tích các yêu cầu (vấn đề logic thực) xác định các biến số đầu vào biểu diễn đối tượng trạng thái (nguyên nhân, kích thích), xác định hàm số ở đầu ra (kết quả cần có, tải). Xác định mối quan hệ hàm – biến là loại quan hệ logic gì.
- Liệt kê bảng chân lý giữa các trạng thái đối tượng (biến) và qua quan hệ logic xác định trạng thái hàm kết quả, thay giá trị 0,1 cho trạng thái hàm và biến (điều này do người thiết kế quy định).
- Tiến hành tối thiểu hoá hàm ra đã có.
- Vẽ sơ đồ logic từ các cổng cơ bản.

Ví dụ 1 : Cần thiết kế yêu cầu của một mạch điện sau: gồm 2 công tắc nối tiếp A và B, 1 công tắc C đấu song song với A và B tới 1 đèn thấp sáng Z (hình 3.1).



Hình 3.1

Ta liệt kê bảng chức năng (bảng 3.1a), sau đó gán cho các trạng thái của A,B,C trị 0 khi ngắt, trị 1 khi nối được bảng chân lý (xem bảng 3.1b). Từ bảng 3.1b có hàm:

$$Z = \bar{A} \bar{B} C + \bar{A} B C + A \bar{B} C + A B \bar{C} + ABC$$

Bảng 3.1a

Biến logic			Hàm logic
Công tắc A	Công tắc B	Công tắc C	Bóng đèn Z
Ngắt	Ngắt	Ngắt	Tắt
Ngắt	Ngắt	Nối	Sáng
Ngắt	Nối	Ngắt	Tắt
Ngắt	Nối	Nối	Sáng
Nối	Ngắt	Ngắt	Tắt
Nối	Ngắt	Nối	Sáng
Nối	Nối	Ngắt	Sáng
Nối	Nối	Nối	Sáng



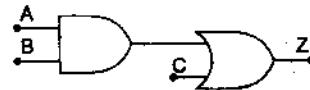
Bảng 3.1b

Biến			Hàm
A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Bìa các nơ của Z và hàm Z sau khi tối thiểu, cấu trúc thực hiện Z cho trên hình 3.2.

$$Z = AB + C$$

C	AB			
	00	01	11	10
0			1	
1	1	1	1	1



Hình 3.2

Ví dụ 2 : Ta hãy xét bài toán khi chỉ có công tắc A nối tiếp công tắc B điều khiển đèn Z.

Sau khi liệt kê các trạng thái và lập bảng chân lý ta có:

Trường hợp 1: ta có bảng 3.2b khi:

Trạng thái ngắt và tắt biểu diễn bởi giá trị 0.

Trạng thái nối và sáng biểu thị bằng giá trị 1.

Trường hợp 2 : ta nhận được bảng 3.2c khi:

0 biểu thị nối và sáng.

1 biểu thị ngắt và tắt.

Trường hợp 3: ta nhận được bảng 3.2d khi:

0 biểu thị ngắt và sáng.

1 biểu thị nối và tắt.

Trường hợp 4: ta có bảng 3.2e khi:

0 biểu thị nối và tắt.

1 biểu thị ngắt và sáng.

Kết quả là quan hệ hàm Z có thể là Z_1 hoặc Z_2 hoặc Z_3 hoặc Z_4 tùy thuộc bài toán ta đã gán cho các trạng thái nối, ngắt, sáng, tắt các giá trị 0 và 1 quy ước khác nhau. Thường phải xuất phát từ trạng thái tích cực (cần quan tâm) của Z là cao hay thấp để quy định cho hợp mục đích.

Bảng 3.2. BẢNG CHÂN LÝ CỦA HÀM RA Z

Công tắc A	Công tắc B	Bóng Z
Ngắt	Ngắt	Tắt
Ngắt	Nối	Tắt
Nối	Ngắt	Tắt
Nối	Nối	Sáng

a)

A	B	Z_1
0	0	0
0	1	0
1	0	0
1	1	1

b) $Z_1 = A.B$

A	B	Z_2
1	1	1
1	0	1
0	1	1
0	0	0

c) $Z_2 = A+B$

A	B	Z_3
0	0	1
0	1	1
1	0	1
1	1	0

d) $Z_3 = \overline{A.B}$

A	B	Z_4
1	1	0
1	0	0
0	1	0
0	0	1

b) $Z_4 = \overline{A+B}$

3.2. CÁC BỘ MÃ HOÁ

3.2.1. Khái niệm. Mã hoá là quá trình dùng văn tự hay ký hiệu để biểu thị một đối tượng. Nếu dùng mã nhị phân để biểu thị đối tượng (là một tín hiệu) ta có quá trình mã hoá nhị phân. Một ký tự nhị phân có 2 giá trị (hai trạng thái) 0 và 1 tương ứng với việc biểu diễn 2 tín hiệu, hay nếu dùng n ký tự nhị phân (n số tự nhiên nguyên) sẽ biểu diễn được 2^n tín hiệu khác nhau. Điều kiện để mã hoá N tín hiệu là:

$$2^n \geq N \quad (3.1)$$

3.2.2. Bộ mã hoá nhị phân

Để cụ thể ta chọn ví dụ $N = 8 = 2^3$

Như vậy tại tám đầu vào của bộ mã hoá xuất hiện một trong tám ký tự $Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7$ (của một số hệ bát phân, ví dụ giống như trong hệ thập phân là các ký tự 0, 1, 2, 3, 4, 5, 6, 7), ta cần 3 ký tự nhị phân (3 bit) theo hệ thức (3.1) để biểu diễn 8 tín hiệu vào này, nghĩa là bộ mã hoá có 3 đầu ra tương ứng A(LSB), B và C(MSB). Như vậy, bộ mã hoá thực hiện quá trình biến đổi một trong 8 đường thành 3 đường.

– Việc mã hoá yêu cầu ở một thời điểm chỉ có một tín hiệu vào $Y_i = 1$ xuất hiện, không cho phép đồng thời có từ 2 tín hiệu vào có giá trị 1 (quan hệ logic của $Y_0, \dots, Y_7$ là không cùng nhau). Ta có bảng chân lý hay bảng mã hoá (xem bảng 3.3).

Bảng 3.3. BẢNG CHÂN LÝ CỦA BỘ MÃ HOÁ BÁT PHÂN SANG NHỊ PHÂN

a) Hàm ra là nhị phân 8421

b) Hàm ra là mã vòng Gray

Tín hiệu vào	Các hàm ra		
	C (2^2)	B (2^1)	A (2^0)
Y_0 (0)	0	0	0
Y_1 (1)	0	0	1
Y_2 (2)	0	1	0
Y_3 (3)	0	1	1
Y_4 (4)	1	0	0
Y_5 (5)	1	0	1
Y_6 (6)	1	1	0
Y_7 (7)	1	1	1

Tín hiệu vào	Các hàm ra		
	C (2^2)	B (2^1)	A (2^0)
Y_0 (0)	0	0	0
Y_1 (1)	0	0	1
Y_2 (3)	0	1	1
Y_3 (2)	0	1	0
Y_4 (6)	1	1	0
Y_5 (7)	1	1	1
Y_6 (5)	1	0	1
Y_7 (4)	1	0	0

– Việc dùng mã nhị phân 3 bit biểu diễn 8 tín hiệu vào có thể có nhiều cách khác nhau: cách 1 theo số đếm nhị phân tự nhiên 8421, cách 2 là bảng chân lý theo mã vòng Gray (xem bảng 1.3 chương 1)...

– Biểu diễn các hàm ra C, B, A theo các biến Y_i tại lối vào ta có hệ hàm ra theo cách 1, từ bảng 3.3a.

$$C = Y_4 + Y_5 + Y_6 + Y_7 = \overline{Y_4} \cdot \overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_7}$$

$$B = Y_2 + Y_3 + Y_6 + Y_7 = \overline{\overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_6} \cdot \overline{Y_7}} \quad (3.2)$$

$$A = Y_1 + Y_3 + Y_5 + Y_7 = \overline{\overline{Y_1} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_7}}$$

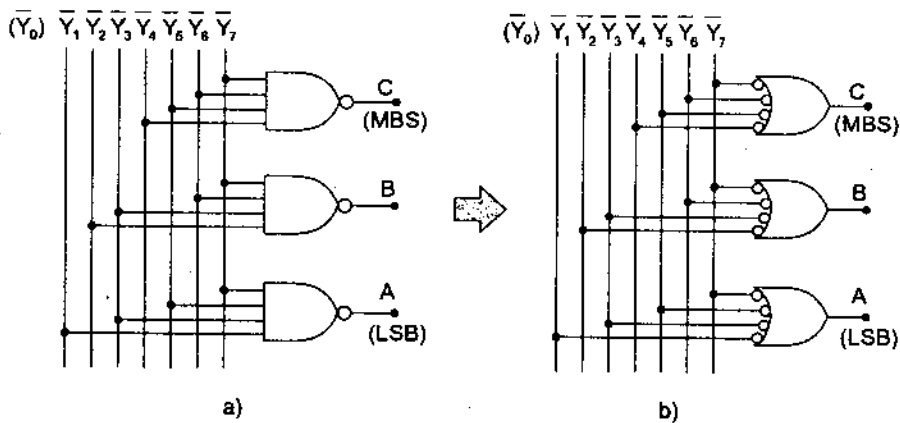
Còn theo cách 2 từ bảng 3.3b ta có:

$$C = Y_4 + Y_5 + Y_6 + Y_7 = \overline{\overline{Y_4} \cdot \overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_7}}$$

$$B = Y_2 + Y_3 + Y_4 + Y_5 = \overline{\overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_4} \cdot \overline{Y_5}} \quad (3.3)$$

$$A = Y_1 + Y_2 + Y_5 + Y_6 = \overline{\overline{Y_1} \cdot \overline{Y_2} \cdot \overline{Y_5} \cdot \overline{Y_6}}$$

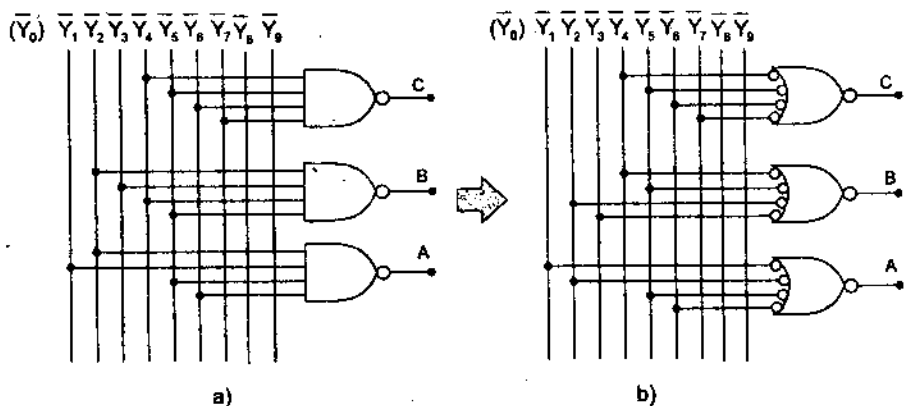
– Từ các kết quả (3.2) xây dựng mạch điện bộ mã hoá nhị phân 3 bit có dạng hình 3.3a hoặc b dùng cổng NAND. Cần lưu ý rằng, trong các sơ đồ này, Y_0 được mã hoá theo quy ước khi $Y_1 = Y_2 = \dots = Y_7 = 0$, trạng thái ra $C = B = A = 0$ là mã của Y_0 .



Hình 3.3. Cấu trúc bộ mã hóa nhị phân 3 bit xây dựng theo bảng chân lý 3.3a. Hệ hàm ra mã 8421

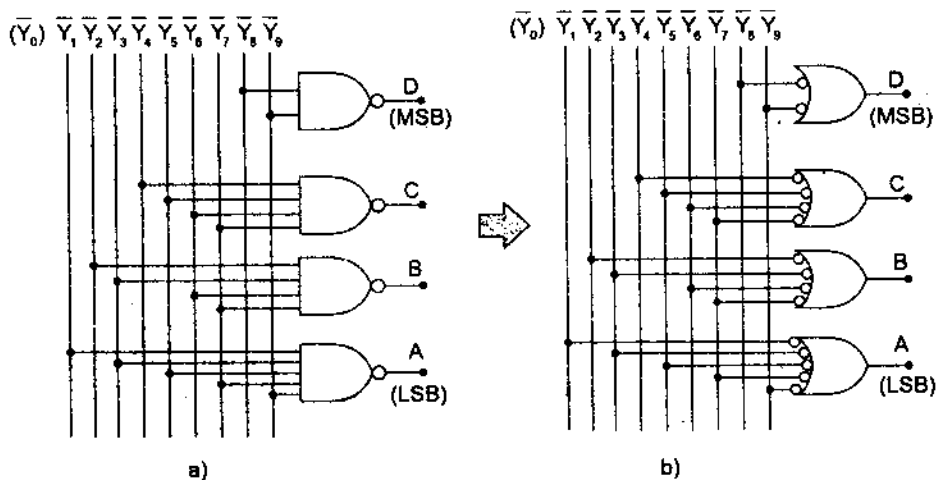
– Để nhấn mạnh các đầu vào yêu cầu các mức tích cực thấp và các đầu ra nhận các mức tích cực cao, cách biểu diễn hình 3.3b cho một mô tả rõ ràng hơn hoạt động của bộ mã hoá (như đã phân tích đối với các ký hiệu logic thay thế tương đương ở chương 2).

– Từ các kết quả (3.3), theo bảng chân lý 3.3b ta nhận được cấu trúc mạch điện bộ mã hoá theo cách 2 với cổng ra là các trị CBA theo mã vòng Gray (hình 3.4).



Hình 3.4. Cấu trúc bộ mã hóa nhị phân 3 bit xây dựng theo bảng chân lý 3.3b. Hệ hàm ra dạng mã vòng Gray.
a) Mạch điện theo ký hiệu chuẩn ;
b) Mạch điện theo ký hiệu tương đương thay thế (hình 2.24)

3.2.3. Bộ mã hoá nhị – thập phân (mã hoá BCD) : là mạch điện chuyển một ký tự thập phân tại đầu vào 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 thành một từ mã biểu diễn số thập phân tương ứng trong hệ nhị phân tại đầu ra.



Hình 3.5. Cấu trúc mạch điện bộ mã hóa BCD 8421
a) Mạch điện theo ký hiệu chuẩn ; b) Mạch điện theo ký hiệu thay thế
Mức ra: tích cực cao ; Mức vào: tích cực thấp.

– Áp dụng công thức (3.1) ta có $n = 4$ tức là phải sử dụng 4 bit nhị phân ABCD biểu diễn các ký tự từ 0 đến 9. Theo bảng 1.2 sẽ có nhiều dạng mã BCD với trọng số khác nhau nên có nhiều cách thực hiện. Ta chọn mã ở đầu

ra là mã BCD 8421, theo đó có bảng trạng thái tương ứng (xem bảng 3.4) và hệ hàm ra có dạng biểu thức (3.4) (chú ý quan hệ $Y_0...Y_9$ là không đồng thời).

$$\begin{aligned} D &= Y_8 + Y_9 = \overline{Y_8} \cdot \overline{Y_9} \\ C &= Y_4 + Y_5 + Y_6 + Y_7 = \overline{Y_4} \cdot \overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_7} \\ B &= Y_2 + Y_3 + Y_6 + Y_7 = \overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_6} \cdot \overline{Y_7} \\ A &= Y_1 + Y_3 + Y_5 + Y_7 + Y_9 = \overline{Y_1} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_7} \cdot \overline{Y_9} \end{aligned} \quad (3.4)$$

- Từ các hệ thức (3.4) ta xây dựng được cấu trúc mạch điện hình 3.5 cho bộ mã hoá BCD8421, chú ý điều kiện đầu vào là các biến $Y_0.....Y_9$ không xuất hiện đồng thời, nếu $Y_i = 1$ thì $Y_j = 0$ ($i \neq j$) và khi $Y_1 = = Y_9 = 0$ thì $A = B = C = D = 0$ ngầm thể hiện $Y_0 = 1$

3.2.4. Bộ mã hoá ưu tiên

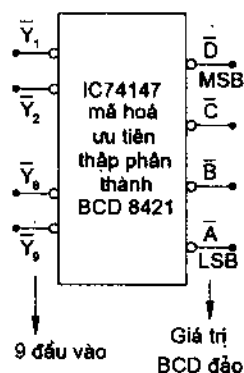
Ở các bộ mã hoá đã xét không tồn tại tình huống có đồng thời từ 2 đầu vào trở lên có trị 1. Thực tế lại gặp phải tình huống có nhiều đầu vào nhận trị 1, lúc này mạch điện chỉ thực hiện mã hoá tín hiệu đầu vào nào có cấp (ưu tiên) cao nhất xét tại lúc chúng cùng tác động. Ta xét với bộ mã hoá ưu tiên 4 bit với 10 đầu vào cho các tín hiệu Y_0 đến Y_9 với tính chất nếu $Y_i = 1$ thì sẽ bỏ qua mọi $Y_j = 1$ khi $j < i$ ($i, j = 0, 1, ..., 9$) tức là chỉ có đầu vào ứng với mức ưu tiên (i) cao nhất được mã hoá (quyết định tới đầu ra) tất cả các đầu vào có $Y_j = 1$ ($j < i$) sẽ không tác động gì tới đầu ra.

IC 74147 thực hiện việc mã hoá ưu tiên có bảng chân lý cho trên bảng (3.4). Vì mạch có 10 đầu vào từ $\overline{Y_0}$ đến $\overline{Y_9}$ và 4 đầu ra $\overline{D}, \overline{C}, \overline{B}, \overline{A}$ tương ứng với các trọng số 8421 của mã BCD tự nhiên, các đầu vào và đầu ra đều có mức tích cực thấp, nghĩa là khi tại các đầu ra xuất hiện mã $\overline{D} = \overline{C} = \overline{B} = \overline{A} = 1$ (ứng với đầu vào $\overline{Y_0} = 0$ hay $Y_0 = 1$) thì từ mã là $D = C = B = A = 0$.

Tương tự như vậy khi đầu vào xuất hiện tín hiệu, ví dụ $Y_7 = 1$ lúc đó tác động tới đầu $\overline{Y_7} = 0$ (mức tích cực thấp) tại các đầu ra theo bảng chân lý (3.4) có tổ hợp $\overline{D} = 1, \overline{C} = \overline{B} = \overline{A} = 0$, tức là ta nhận được giá trị của từ mã $D = 0, C = B = A = 1$. Từ nhận xét đó, mã BCD 8421 tại đầu ra chính là số bù 1 của các giá trị hàm ra ở bảng chân lý 3.4.

Bảng 3.4. BẢNG CHÂN LÝ BỘ MÃ HÓA ƯU TIÊN BCD 8421

Các giá trị biến vào $\bar{Y}_i$										Hàm ra BCD 8421				
$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$	$\bar{Y}_4$	$\bar{Y}_5$	$\bar{Y}_6$	$\bar{Y}_7$	$\bar{Y}_8$	$\bar{Y}_9$	$\bar{D}$	$\bar{C}$	$\bar{B}$	$\bar{A}$	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	(0)
x	x	x	x	x	x	x	x	x	0	0	1	1	0	(9)
x	x	x	x	x	x	x	x	0	1	0	1	1	1	(8)
x	x	x	x	x	x	x	0	1	1	1	0	0	0	(7)
x	x	x	x	x	x	0	1	1	1	1	0	0	1	(6)
x	x	x	x	x	0	1	1	1	1	1	0	1	0	(5)
x	x	x	x	0	1	1	1	1	1	1	0	1	1	(4)
x	x	x	0	1	1	1	1	1	1	1	1	0	0	(3)
x	x	0	1	1	1	1	1	1	1	1	1	0	1	(2)
x	0	1	1	1	1	1	1	1	1	1	1	1	0	(1)

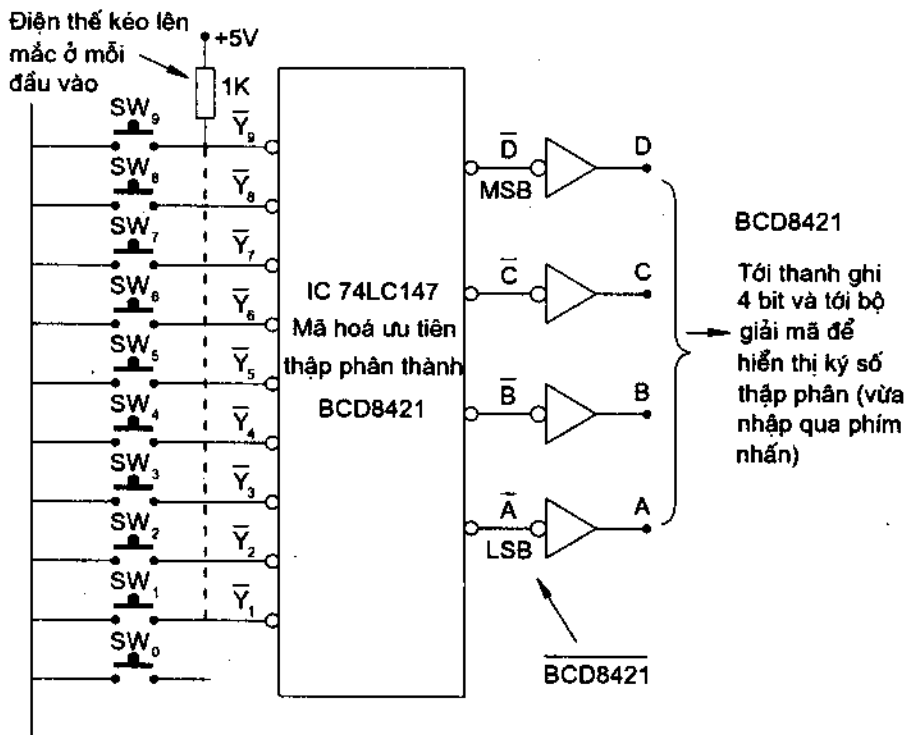


Thực tế khi không có đầu vào nào được kích hoạt, các đầu ra ở mức cao ($\bar{D} = \bar{C} = \bar{B} = \bar{A} = 1$) ứng với trạng thái BCD 8421 là 0 0 0 0, do vậy không cần đầu vào $\bar{Y}_0$ vì đã được tự nhiên công nhận. Có thể chuyển thành mã BCD 8421 nếu nối với mỗi đầu ra một cổng đảo 1/6 IC 4049.

Trong bảng 3.4 các đầu vào cấp thấp ($j < 1$) vì có mức ưu tiên thấp hơn nên không có tác dụng gì tới đầu ra (khi cấp ưu tiên cao đã kích hoạt đồng thời) nên được đánh dấu "x" tức là có thể 0 hay 1.

3.2.5. Bộ mã hoá chuyển mạch (hình 3.6) sử dụng 10 chuyển mạch bàn phím bấm S_i (ví dụ trên máy tính bấm) biểu thị 10 ký số thập phân từ 0 đến 9. Các chuyển mạch SW_i là loại thường hở mạch, các đầu vào $\bar{Y}_0 \div \bar{Y}_9$ được nối với +5V qua các điện trở $1k\Omega$ để kéo mức cao lên lúc SW_i hở mạch. Khi 1 phím SW_i nào đó ($i = 0 \div 9$) được nhấn nối mạch, đầu vào tương ứng chuyển về mức thấp kích hoạt cho cửa $\bar{Y}_i$ và bộ mã hoá làm việc với đầu vào này, tương ứng sẽ cho tại các đầu ra 1 từ mã BCD 8421 biểu diễn số i thập

phân trên bàn phím. Bằng cách này 1 ký số thập phân từ 0 đến 9 đã qua mạch hình 3.6 chuyển thành một biểu diễn BCD 8421 nhập vào, ví dụ một thanh ghi 4 bit nhị phân BCD chuẩn bị việc tính toán số học nhị phân với nó. Nếu một máy tính tay có 8 ký số thập phân, thanh ghi sẽ có 4 byte ghi 32 ký tự biểu diễn 8 số này. Trường hợp nếu nhấn (nhầm) nhiều số một lúc, số phím cao nhất sẽ được ưu tiên mã hoá. Mặt khác các từ mã BCD 8421 đã được ghi theo một cổng song song đưa tới bộ giải mã, cho phép hiện số thập phân (sau khi qua mạch giải mã, xem phần 3.3) trên màn của máy tính của ký số phím ta vừa nhấn.



Hình 3.6. Bộ mã hóa chuyển mạch bàn phím thập phân thành BCD 8421

3.3. CÁC BỘ GIẢI MÃ

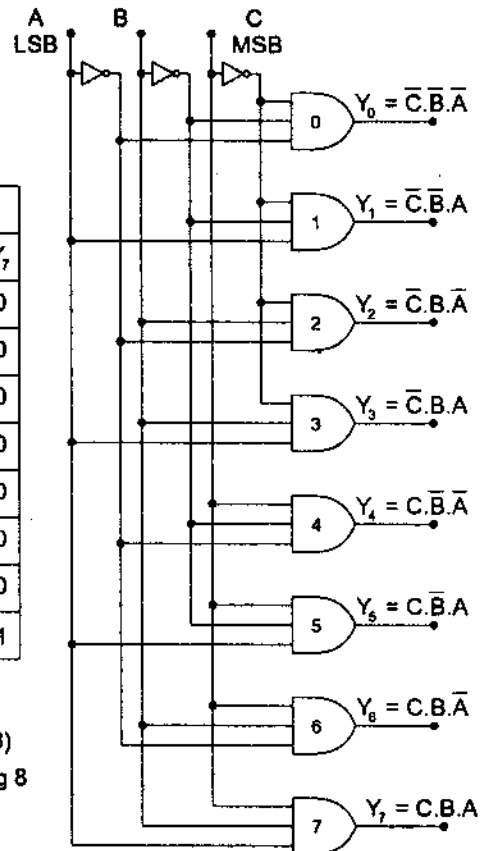
Bộ giải mã có nhiệm vụ chuyển đổi (tiếp nhận) 1 từ mã nhị phân đầu vào (khi chúng tác động đồng thời tới các đầu vào) thành một tín hiệu logic duy nhất ở một đầu ra nào đó tương ứng với mã nhị phân đã tác động. Như vậy với N đầu vào có thể nhận giá trị 0 hay 1, sẽ có 2^N tổ hợp mã đầu vào. Với mỗi tổ hợp mã đầu vào chỉ có một đầu ra tương ứng với biểu diễn của mã đầu vào được kích hoạt lên mức tích cực, các đầu ra còn lại ở mức đối lập với mức tích cực. Khi không dùng hết các tổ hợp mã đầu vào, bộ giải mã được thiết kế loại bỏ các tổ hợp này để chúng không tác động tới bất kỳ đầu ra nào.

3.3.1. Bộ giải mã nhị phân – bát phân (3 đường thành 8 đường – 1 trong 8)

1. *Bảng chân lý và mạch điện thực hiện việc giải mã 3 đường thành 8 đường (hình 3.7)*

Đầu vào			Đầu ra							
C MSB	B	A LSB	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

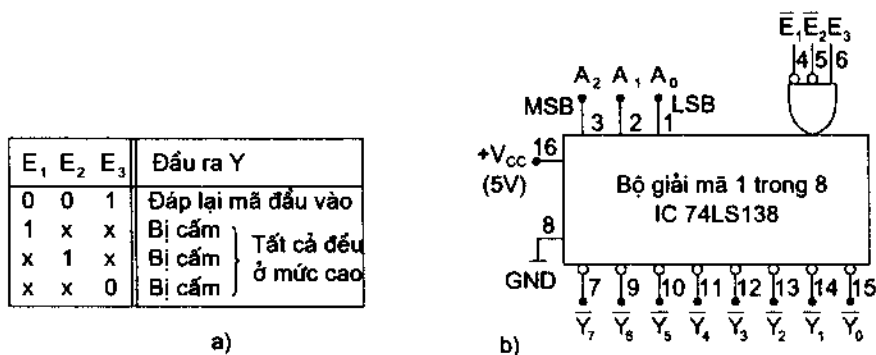
- a) Bảng chân lý bộ giải mã
3 đường thành 8 đường (1 trong 8)
b) Mạch điện của bộ giải mã 1 trong 8



Hình 3.7. Bộ giải mã nhị phân – bát phân (1 trong 8 hay 3 đường thành 8 đường)

Với 3 đầu vào C, B, A sẽ xuất hiện $2^3 = 8$ tổ hợp khác nhau của các trạng thái vào, khi đó qua các cổng AND từ 0 tới 7. Chỉ một trong các tổ hợp tác động cùng lúc ở các đầu vào có gây phản ứng tích cực ở đầu ra tương ứng của một trong các cổng AND này, tức là đầu ra này lên mức tích cực. Ví dụ với tổ hợp mã vào là 101 ($= C\bar{B}A$) chỉ duy nhất cổng AND₅ chuyển đầu ra lên mức tích cực cao.

2. Bộ giải mã 1 trong 8 có đầu vào cho phép E (ENABLE) dùng để điều khiển hoạt động (được phép hay không được phép) của nó. Hình 3.8 là một bộ giải mã dạng có đầu vào cho phép dùng IC 74LS138. Tuy nhiên có thể sử dụng mạch hình 3.7 đã xét với việc sử dụng các cổng AND 4 đầu vào, đầu vào thứ tư luôn ở mức cao ($E = 1$) là được phép hoạt động, còn khi $E = 0$ đưa tới đầu vào thứ tư thì tất cả các cổng AND từ 0 đến 7 đều bị cấm hoạt động (không cho phép) và bộ giải mã sẽ không làm việc với mọi tổ hợp mã vào.



Hình 3.8. Bảng chân lý đầu vào cho phép và ký hiệu logic của IC 74LS138 giải mã 1 trong 8.

a) Bảng chân lý;

Đầu vào E_1, E_2 mức tích cực thấp; Đầu vào E_3 mức tích cực cao

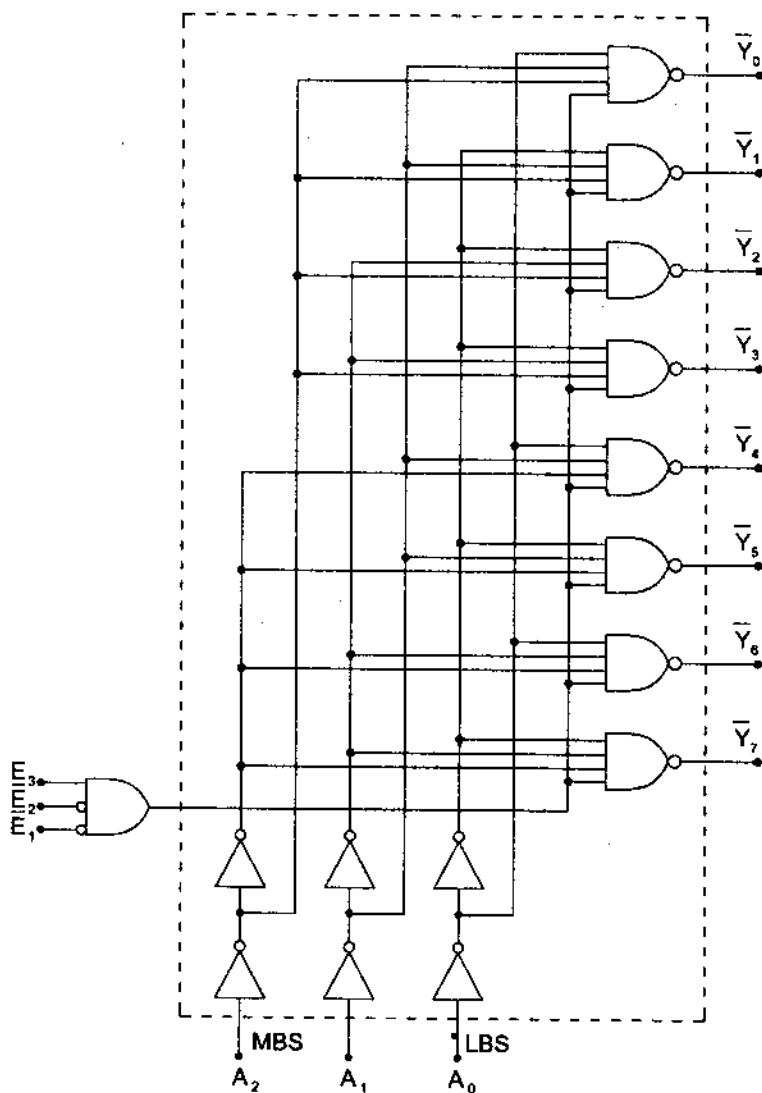
b) Ký hiệu logic của IC 74LS138;

Đầu vào A_2, A_1, A_0 mức tích cực cao; Đầu ra $\bar{Y}_i$ mức tích cực thấp.

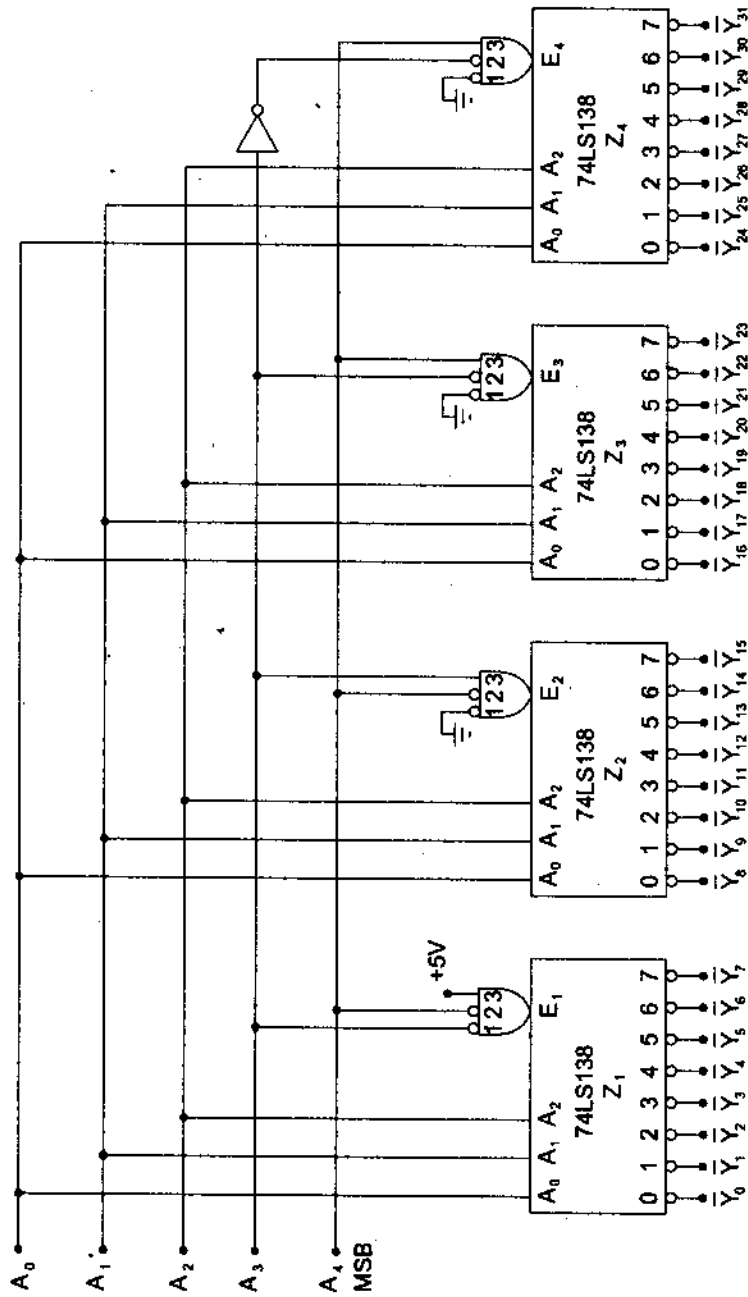
Từ bảng chân lý 3.8a chỉ khi $\bar{E}_1 = \bar{E}_2 = 0$ và $E_3 = 1$ bộ giải mã mới được phép hoạt động để kích hoạt một đầu ra $\bar{Y}_i$ đến mức tích cực cao (hay chính xác hơn $\bar{Y}_i = 0$; $i = 0, 7$). Các trường hợp còn lại với mọi tổ hợp $\bar{E}_1, \bar{E}_2, E_3$ các đầu vào tín hiệu A_2, A_1, A_0 đều bị cấm, bộ giải mã không được phép hoạt động.

- Việc sử dụng 2 cổng đảo liên tiếp (xem hình 3.8c) mục đích để sửa dạng xung tín hiệu (biểu diễn mã vào) cho sườn xung thẳng hơn đối với A_2, A_1, A_0 , trong đó 3 cổng dùng để tạo các mức thấp $\bar{A}_2, \bar{A}_1, \bar{A}_0$ của tín hiệu vào.

- Có thể sử dụng 4 IC74LS138 kết hợp với các đầu vào cho phép E để tạo thành bộ giải mã 1 trong 32 thường gặp trong thực tế như hình 3.9 biểu thị, khi đó tín hiệu vào có 5 bit nhị phân A_4, A_3, A_2, A_1, A_0 . Bốn bộ giải mã 1 trong 8 đã xét là Z_1, Z_2, Z_3, Z_4 có mỗi bộ 8 đầu ra kết hợp lại thành 32 đầu ra : từ $\bar{Y}_0$ đến $\bar{Y}_{31}$.



Hình 3.8c. Mạch điện bộ giải mã 1 trong 8 dùng IC 74LS138 có đầu vào cho phép E



Hình 3.9. Bốn bộ giải mã 1 trong 8 tạo thành một bộ giải mã 1 trong 32

Mỗi từ mã đầu vào 5 bit $A_4 A_3 A_2 A_1 A_0$ sẽ điều khiển để chỉ một trong 32 đầu ra ở mức tích cực thấp. Các mã đầu vào được phân nhóm theo trọng số, sử dụng bit A_4 (2^4) và bit A_3 (2^3) để tạo mã địa chỉ đưa tới các đầu vào cho phép E nhằm buộc các bộ giải mã 1 trong 8 từ Z_4 tới Z_1 chỉ được phép hoạt động theo từng khối nhóm mã đầu vào có trọng số từ cao nhất (1 1 1 1 1) đến trọng số thấp nhất (0 0 0 0 0). Ví dụ khi đầu vào cho phép $E_4 = 001$. Theo bảng hình 3.8a chỉ duy nhất Z_4 làm việc với các tổ hợp mã từ 1 1 1 1 1 đến 1 1 0 0 0, vậy nhãn địa chỉ cho phép của Z_4 là $A_4 A_3 = 11$; của Z_3 là 10, của Z_2 là 01 và của Z_1 là 00, Z_1 chỉ được cho phép kích hoạt khi các tổ hợp mã đầu vào thuộc nhóm 00 111 đến 0 0 0 0 0.

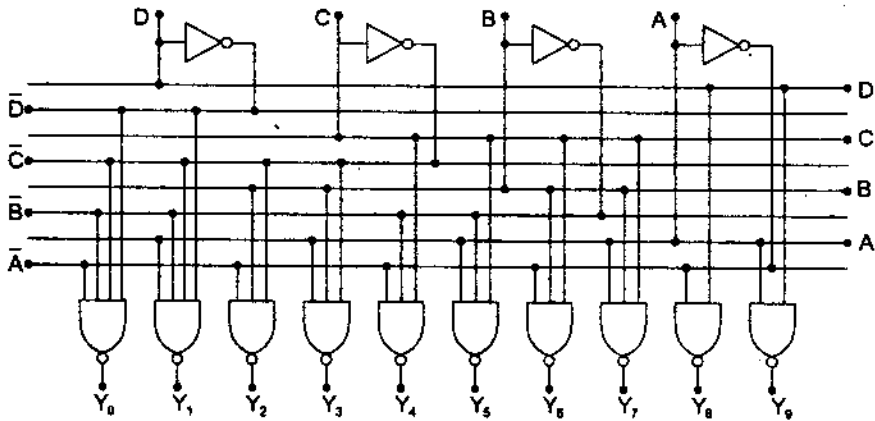
3.3.2. Bộ giải mã 1 trong 10 (biến đổi nhị phân BCD thành thập phân)

1. Nguyên lý hoạt động của bộ giải mã 1 trong 10 là khi xuất hiện 1 tổ hợp mã đầu vào tác động, chỉ một trong mười đầu ra của bộ giải mã sẽ chuyển đến trạng thái tích cực biểu diễn số thập phân tương ứng với từ mã đầu vào này.

Bảng chân lý và mạch điện của bộ giải mã 1 trong 10 được cho trên bảng 3.5 và hình 3.10.

Bảng 3.5. BẢNG CHÂN LÝ BỘ GIẢI MÃ 1 TRONG 10

Mã đầu vào				Hàm ra									
D	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9
0	0	0	0	0									
0	0	0	1		0								
0	0	1	0			0				1			
0	0	1	1				0						
0	1	0	0					0					
0	1	0	1						0				
0	1	1	0							0			
1	1	1	1				1				0		
1	0	0	0									0	
1	0	0	1										0
1	0	1	0										
1	0	1	1										
1	1	0	0										
1	1	0	1										
1	1	1	0										
1	1	1	1										

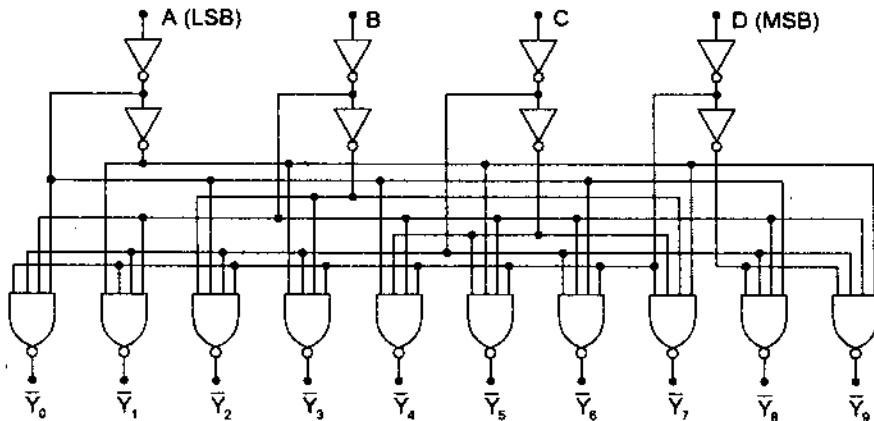


Hình 3.10. Bộ giải mã 1 trong 10: Mạch điện nguyên lý

Bảng trạng thái 3.5 được viết với các tổ hợp mã BCD 8421 4bit trong đó 6 trạng thái cuối cùng không sử dụng đến (chúng không xuất hiện trong tình huống bình thường của bộ giải mã) và được đánh dấu x trong bảng chân lý. Thực hiện việc lập bảng các nơ cho hệ các hàm ra Y_0 đến Y_9 , thực hiện việc rút gọn (có sử dụng các tổ hợp mã không dùng đến), ta nhận được hệ hàm ra $\bar{Y}_0$ đến $\bar{Y}_9$ có dạng hệ thức (3.5):

$$\begin{aligned}
 \bar{Y}_9 &= DA; & Y_9 &= \overline{DA} & \bar{Y}_8 &= D\bar{A}; & Y_8 &= \overline{D\bar{A}} \\
 \bar{Y}_7 &= CBA; & Y_7 &= \overline{CBA} & \bar{Y}_6 &= C\bar{B}\bar{A}; & Y_6 &= \overline{C\bar{B}\bar{A}} \\
 \bar{Y}_5 &= C\bar{B}A; & Y_5 &= \overline{C\bar{B}A} & \bar{Y}_4 &= C\bar{B}\bar{A}; & Y_4 &= \overline{C\bar{B}\bar{A}} \\
 \bar{Y}_3 &= \bar{C}BA; & Y_3 &= \overline{\bar{C}BA} & \bar{Y}_2 &= \bar{C}\bar{B}\bar{A}; & Y_2 &= \overline{\bar{C}\bar{B}\bar{A}} \\
 \bar{Y}_1 &= \bar{D}\bar{C}\bar{B}A; & Y_1 &= \overline{\bar{D}\bar{C}\bar{B}A} & \bar{Y}_0 &= \bar{D}\bar{C}\bar{B}\bar{A}; & Y_0 &= \overline{\bar{D}\bar{C}\bar{B}\bar{A}}
 \end{aligned} \quad (3.5)$$

Tương ứng với các hàm ra (3.5) ta nhận được mạch điện hình 3.10 cho bộ giải mã 1 trong 10 gồm 4 cổng đảo và 10 cổng NAND. Hình 3.11 là mạch điện cấu trúc của IC loại MSI 7442 (hay 74 LS42/74HC42) thực hiện việc giải mã 1 trong 10, có đặc điểm khác là sử dụng 2 cổng đảo với mỗi đầu vào để sửa sườn thẳng đứng của xung tín hiệu cho tốt hơn.



Hình 3.11. Mạch điện logic IC 7442/74LS42/74HC42 bộ giải mã BCD 8421 thành thập phân

– Chú ý IC giải mã 1 trong 10 7445 là loại TTL, có đầu ra dạng hở colectơ nhờ đó có khả năng thu nhận tới 80 mA ở đầu ra khi ở tích cực thấp và có thể kéo lên tới +30V khi ở tích cực cao nhờ đó phù hợp với việc trực tiếp ghép tải là LED chỉ thị hay role hoặc động cơ 1 chiều có công suất nhỏ.

– Khi chỉ dùng 3 đầu vào tín hiệu CBA, IC 7442 có chức năng như 1 bộ giải mã 1 trong 8, đầu vào thứ tư D sử dụng làm đầu vào cho phép. Khi đó đầu ra $\bar{Y}_8$ và $\bar{Y}_9$ luôn ở mức cao.

3.3.3. Bộ giải mã từ BCD thành 7 đoạn (LED) hiển thị

1. Màn hình LED được sử dụng một cách phổ biến nhất để hiển thị sáng các ký số thập phân từ 0 tới 9 và đôi khi các ký tự từ A tới F. Cấu tạo màn hình gồm 7 thanh LED độc lập nhau được bố trí như trên hình 3.12 với 2 loại điển hình : loại anốt chung (khi tất cả 7 anốt của các điốt LED nối chung với nhau) và loại katốt chung (khi 7 katốt của chúng đấu chung nhau).

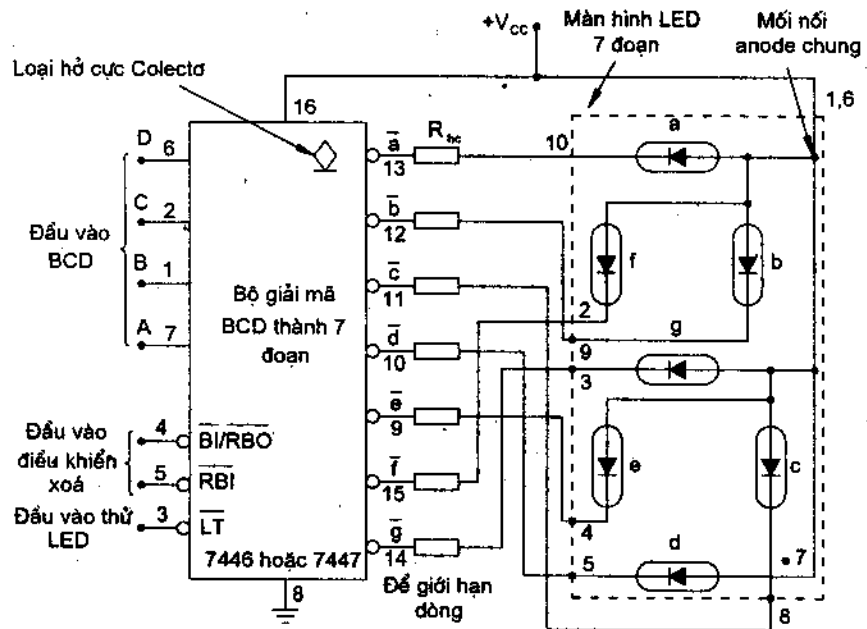
– Khi sử dụng anốt chung được nối tới 5V, 1 điện trở hạn chế dòng điện ở mức từ 5mA tới 40mA cho mỗi đoạn LED từ katốt của mỗi đoạn sẽ nối tới đầu ra tương ứng của bộ giải mã có mức tích cực thấp (mức 0) và cấu trúc loại hở cực colectơ để chịu dòng hút lớn.

– Với loại có katốt chung thì nhóm katốt chung được nối tới 0V qua điện trở hạn dòng (vài trăm Ω), anốt của mỗi thanh LED được nối tới đầu ra của bộ giải mã có mức tích cực cao qua một bộ đệm (mức 1). Khi thỏa mãn điều kiện của mức tích cực LED sẽ phát sáng vì có dòng chảy qua và hiển thị vạch mà nó biểu diễn.

– Các IC giải mã trong trường hợp dùng LED katốt chung không được nối trực tiếp tới các thanh LED vì đầu ra của TTL hay CMOS không đủ

dòng phun cung cấp cho LED khi đó cần giao diện thường là tranzitor BJT đệm giữa IC giải mã và thanh LED có katốt nối chung.

– Các ký tự của các đoạn LED được ký hiệu lần lượt là a, b, c, d, e, f, và g (hình 3.12).



Hình 3.12. IC giải mã 7446/ 7447 (BCD thành 7 đoạn) điều khiển màn hình LED 7 đoạn có anốt chung (FND 350/360/310)

2. Bộ giải mã BCD thành 7 đoạn có 4 đầu vào tín hiệu DCBA dạng mã BCD8421, 7 đầu ra mức tích cực thấp là $\bar{a}$, $\bar{b}$, $\bar{c}$, $\bar{d}$, $\bar{e}$, $\bar{f}$ và $\bar{g}$, ngoài ra nó còn có các đầu vào điều khiển và đầu vào thử LED. Tương ứng với mỗi tổ hợp mã vào BCD trong mười tổ hợp đầu tiên (biểu diễn các ký số từ 0 đến 9) trong hệ 2 một nhóm đầu ra sẽ được chuyển đến mức tích cực thấp và các đoạn LED ứng với các đầu ra này sẽ sáng để phối hợp và hiển thị con số thập phân tương ứng của từ mã vào đã tác động. Bảng chân lý của bộ giải mã BCD thành 7 đoạn cho trên bảng 3.6.

Khi xét từng hàm ra a, b ..., f, g ta được hệ hàm ra sau khi đã tối thiểu là :

$$\begin{aligned}
 \bar{a} &= \overline{D+B+CA+CA} & \bar{e} &= \overline{CA+BA} \\
 \bar{b} &= \overline{C+BA+BA} & \bar{f} &= \overline{D+CB+CA+BA} \\
 \bar{c} &= \overline{C+B+A} & \bar{g} &= \overline{D+CB+CB+BA} \\
 \bar{d} &= \overline{D+CB+BA+CA+CBA}
 \end{aligned} \quad (3.6)$$

Bảng 3.6. BẢNG GIÁ TRỊ (CHÂN LÝ) CỦA BỘ GIẢI MÃ BCD THÀNH 7 ĐOẠN

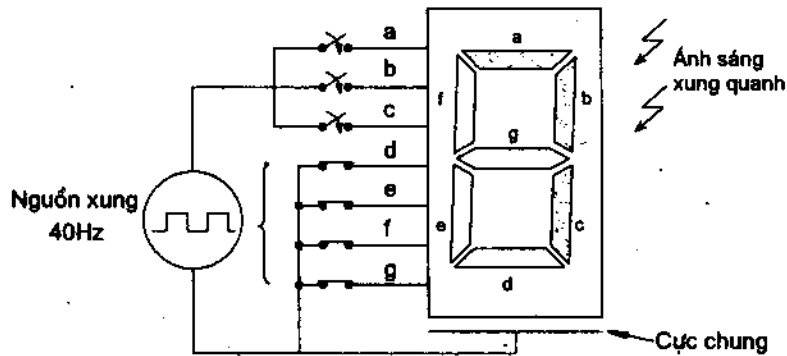
Mã BCD đầu vào				Đầu ra mã 7 đoạn							Thập phân được hiển thị
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	0	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	1	0	0	9

Việc thực hiện các hàm (3.6) nhờ các cổng NOR – AND đã giới thiệu ở chương 2 được cấu trúc trong chip IC giải mã 7446 hay 7447 với cực collector của tranzitor tăng ra để hở tạo khả năng chịu dòng hút lớn từ các đoạn LED. Ví dụ với ký số 8 thập phân, cả 7 đoạn LED đều sáng, mỗi đoạn có dòng là 10mA thì tranzitor tăng ra cần chịu được dòng hút là 10mA. 7 đoạn = 70mA. Với điện thế trên LED lúc sáng là 2,7V thì điện trở hạn chế dòng trong mạch hình 3.12 cần chọn giá trị là:

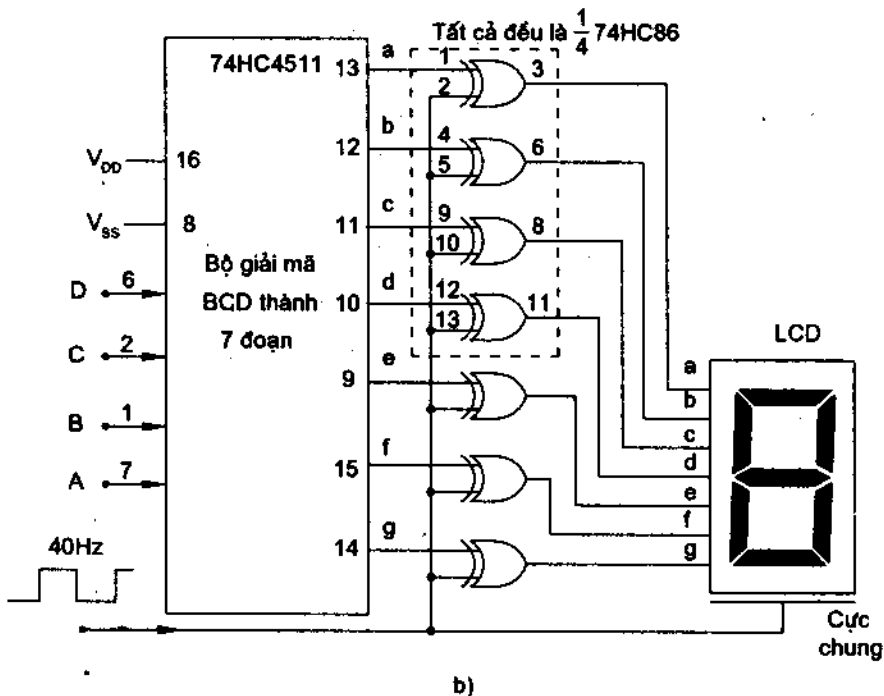
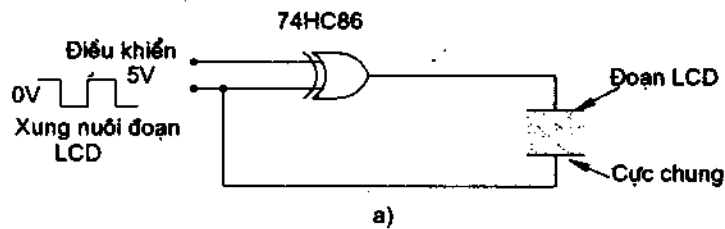
$$R_{\text{hạn chế}} = \frac{5V - 2,7V}{10mA} = 230\Omega ; \text{ ta chọn giá trị chuẩn là } 220\Omega$$

3. Màn hình tinh thể lỏng LCD và phương pháp kích thích sáng màn LCD

– Màn hình LCD được bố trí kiểu 7 đoạn như LED làm việc với điện thế xoay chiều (25Hz tới 60 Hz) với dòng tiêu thụ rất nhỏ hơn so với LED, đây là ưu điểm quan trọng đặc biệt với các thiết bị chạy bằng pin nhỏ (như máy tính bấm, máy tính xách tay...) bên cạnh nhược điểm là độ sáng yếu vì LCD tận dụng ánh sáng có sẵn (nguồn tự nhiên của môi trường hay nguồn sáng nhỏ chiếu hậu) và điều khiển phản xạ của ánh sáng này. Hình 3.13 mô tả 1 màn hình LCD gồm 7 vạch (a, b, c, d, e, f, g giống LED) với 1 điện cực chung và các đầu vào tín hiệu từ a tới g tương ứng ; khi a, b, c được kích thích bởi 1 nguồn xung biên độ $\pm 5V$ tần số 40Hz chúng sẽ phản xạ sáng, các đầu vào còn lại do không được điện thế xoay chiều tác động nên vẫn tối, số 7 được hiển thị trên màn hình LCD.



Hình 3.13. Màn hình LCD và phương pháp kích thích sáng số 7 thập phân



Hình 3.14. a) Phương pháp kích thích một đoạn LCD (khi đầu vào điều khiển ở mức cao, cổng logic XOR được nối mạch);
b) Mạch điện: bộ giải mã BCD 8421 thành 7 đoạn kích thích tải là LCD hiển thị số thập phân tương ứng.

– Mạch điện hình 3.14b sử dụng 1 IC giải mã BCD thành 7 đoạn 74 HC4511 và 7 cổng hàm khác nhau EX-OR loại CMOS 74 HC86 để điều khiển trạng thái nối/ngắt của đoạn LCD tương ứng bằng cách các mức ra của bộ giải mã (a, b, c, d, e, f, g) cho phép ở mức tích cực hay không cho phép xung kích thích 40Hz vào các đoạn LCD để biểu thị ký số thập phân tương ứng. Cách điều khiển này thể hiện trên hình 3.14a, khi đầu vào điều khiển của cổng 74 HC86 ngắt không cho phép xung vuông qua và vạch tương ứng sẽ tối (trong ví dụ hình 3.13, chỉ a, b, c có giá trị để cổng 74 HC86 được phép mở).

– Bằng cách tương tự với màn LCD là các ma trận ví dụ 640 (cột) $\times$ 480 (hàng) cỡ lớn với 2400 mối nối với LCD (gọi là các điểm ảnh) đi qua các bộ lọc màu tạo 3 màu cơ bản (đỏ, xanh lục, xanh da trời) tạo màu cho từng điểm ảnh với một vi điện tử loại cỡ lớn VLSI điều khiển có thể quét màn hình LCD với tốc độ cao, thiết lập nên các mẫu ký số, ký tự và thậm chí các ảnh video chất lượng cao.

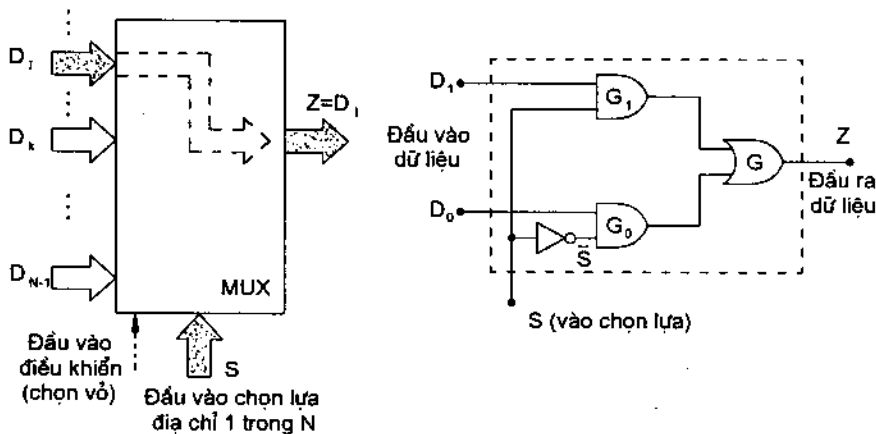
3.4. BỘ CHỌN DỮ LIỆU (BỘ ĐÓN KÊNH – MUX)

3.4.1. Bộ đón kênh (Multiplexer – MUX) theo định nghĩa tổng quát là một thiết bị tổ hợp có khả năng chọn một trong các tín hiệu đặt tới đầu vào theo mức địa chỉ của nó truyền tới đầu ra duy nhất. Bộ MUX số là mạch logic tổ hợp chấp nhận cùng một lúc nhiều đầu vào ở dạng dữ liệu số, chọn ra một trong các dữ liệu này tại một thời điểm xác định để chuyển tới đầu ra. Như vậy hoạt động của MUX chọn đầu vào dữ liệu là do một hay một nhóm đầu vào đặc biệt được gọi là đầu vào chọn lựa (Select-S) quyết định (xem hình 3.15a). Vì vậy, đầu vào S là một tín hiệu thể hiện địa chỉ được chọn của một trong số các đầu vào, với N đầu vào dữ liệu truy nhập đồng thời tới MUX cần tạo ra các địa chỉ khác nhau bằng cách sử dụng n ký số nhị phân (n bit) tuân theo điều kiện $2^n \geq N$, ví dụ với $N = 20$ thì $n = 5$; $N = 16$ thì $n = 4$; $N = 8$ thì $n = 3$... $N = 4$ thì $n = 2$ và khi $N = 2$ thì chọn $n = 1$.

3.4.2. Mạch điện các bộ đón kênh

1. Bộ đón kênh 2 đầu vào ($N = 2$) cho trên hình 3.15b hoạt động theo mức logic của S : khi $S = 0$ thì G_0 mở và $Z = D_0$, còn khi $S = 1$ cổng G_1 sẽ mở cho phép D_1 qua và $Z = D_1$.

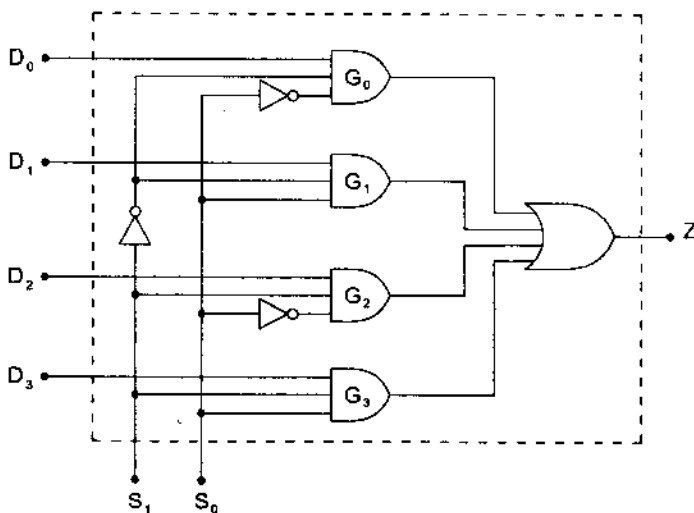
2. Mạch điện một bộ đón kênh 4 đầu vào ($N = 4$) được cho trên hình 3.16 mã địa chỉ chọn đầu vào được thực hiện nhờ 2 bit nhị phân S_1 và S_0 với 4 tổ hợp của chúng sẽ tạo được mã cho 4 địa chỉ đầu vào dữ liệu tương ứng là



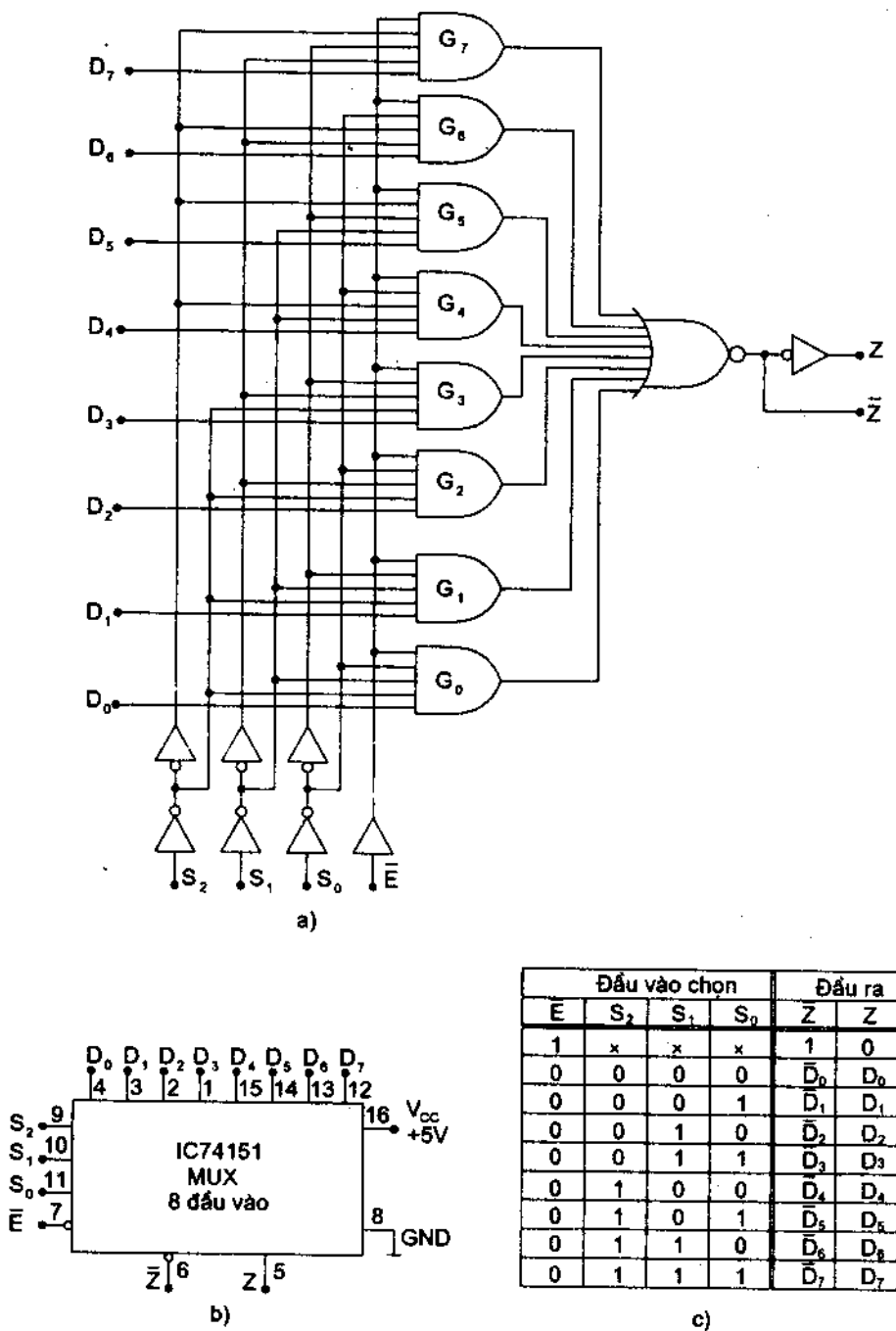
Hình 3.15

- a) Sơ đồ chức năng của MUX; b) Bộ dẫn kênh 2 đầu vào dữ liệu.

$D_0 (S_1 = S_0 = 0)$, $D_1 (S_1 = 0, S_0 = 1)$, $D_2 (S_1 = 1, S_0 = 0)$ và $D_3 (S_1 = S_0 = 1)$ qua các cổng AND, các tổ hợp S_0, S_1 tác động tới 2 cổng điều khiển việc chọn, cổng thứ 3 dành cho dữ liệu D_i ($i = \overline{0,3}$). Ví dụ với tổ hợp $S_1 S_0 = 11$. Chỉ có cổng G_3 được chọn cho phép D_3 truyền qua MUX tới đầu ra, ta có $Z = D_3$, tương tự $Z = D_1$ khi có địa chỉ $S_1 S_0 = 01$.

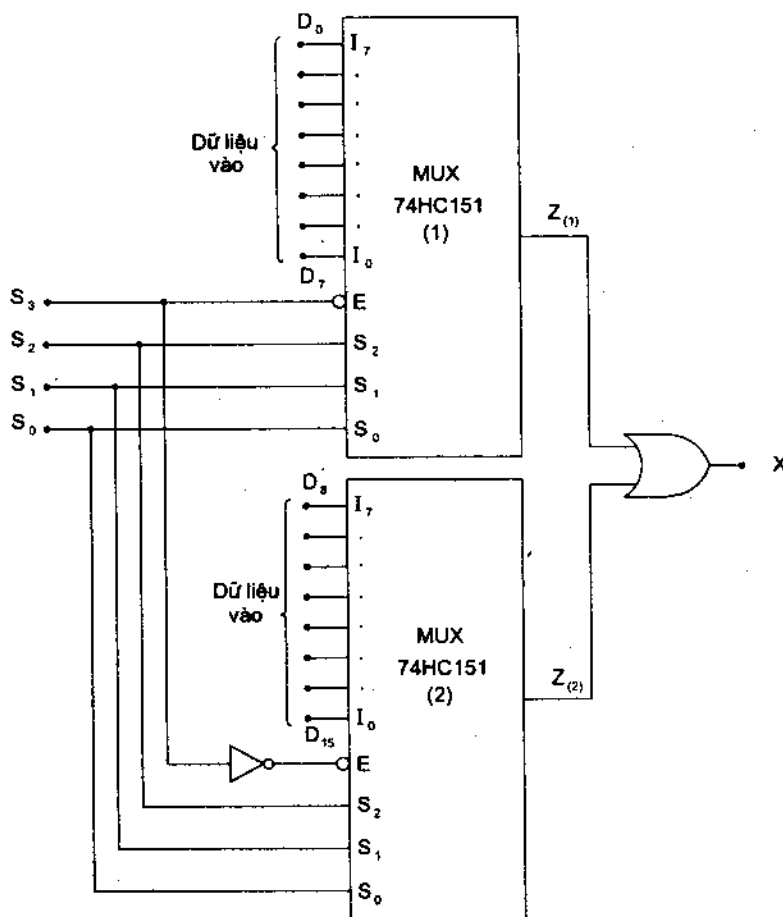


Hình 3.16. Bộ MUX 4 đầu vào



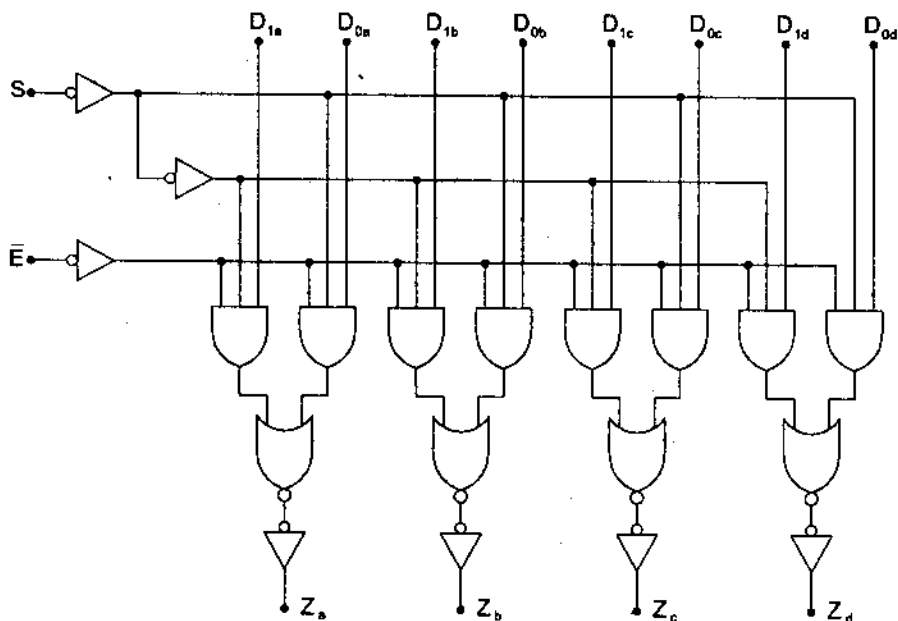
Hình 3.17. a) Mạch điện nguyên lý của IC MUX 74151, 74LS151, 74HC151 (chọn 1 trong 8);
b) Ký hiệu logic của IC giải mã 8 đầu vào 74151;
c) Bảng chân lý của IC giải mã 8 đầu vào 74151.

3. Mạch điện bộ dồn kênh 8 đầu vào ($N = 8$) kết cấu trong 1 chip IC 74151 (hoặc 74LS151, 74HC151) cho trên hình 3.17. Ngoài 3 bit S_1, S_2, S_0 xuất hiện ở các đầu vào chọn để tạo mã địa chỉ cho các đầu vào dữ liệu tương ứng từ D_0 đến D_7 với các tổ hợp từ 000 đến 111, vi điện tử MUX loại này còn có thêm 1 đầu vào cho phép $\bar{E}$ ở mức tích cực thấp, cung cấp hai đầu ra Z và $\bar{Z}$: khi $\bar{E} = 0$ đầu vào được chọn sẽ truyền tới đầu ra $\bar{Z}$ (tích cực cao) và đầu ra Z (tích cực thấp) đồng thời, còn khi $\bar{E} = 1$ thì $Z = 0$ bất chấp mọi trạng thái của các đầu vào dữ liệu (MUX lúc này bị cấm, không cho hoạt động).

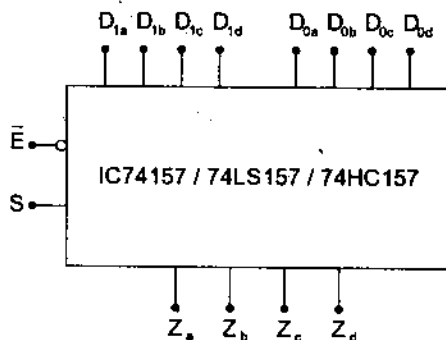


Hình 3.18. MUX 16 đầu vào dữ liệu và 4 đầu vào chọn dùng 2 IC 74HC151

4. Khi cần 1 bộ MUX có 16 đầu vào dữ liệu từ D_0 đến D_{15} , ta có thể sử dụng 2 IC dồn kênh 8 đầu vào. Ví dụ 74 HC151 kết hợp với 4 bit chọn địa



a)



b)

Vào chọn		Ra			
$\bar{E}$	S	Z_a	Z_b	Z_c	Z_d
1	X	0	0	0	0
0	0	D_{0a}	D_{0b}	D_{0c}	D_{0d}
0	1	D_{1a}	D_{1b}	D_{1c}	D_{1d}

c)

Hình 3.19. IC dẫn kênh chấp bốn 2 đầu vào 74157

a) Mạch điện logic IC 74157 chấp 4 MUX 2 đầu vào;

b) Ký hiệu logic IC chấp 4 2 đầu vào MUX 74157/74LS157/ 74HC157;

c) Bảng chân lý của IC 74157.

chỉ S_3, S_2, S_1, S_0 như hình 3.18 thể hiện. Đầu vào chọn địa chỉ S_3 sẽ quyết định IC dẫn kênh nào được phép hoạt động vì với cả hai IC S_3 được nối tới đầu vào $\bar{E}$ trực tiếp (với nhóm trên) hay qua cổng NOT (với nhóm dưới). Khi $S_3 = \bar{E}_{(1)} = 0$, MUX (1) sẽ được phép làm việc với bộ mã địa chỉ $S_2 S_1 S_0$ cho đầu ra $Z_{(1)}$ tương ứng với một trong các dữ liệu D_0 đến D_7 , lúc này $\bar{S}_3 = 1 = \bar{E}_{(2)}$ nên MUX(2) bị cấm hoạt động. Khi $S_3 = \bar{E}_{(1)} = 1$, tình hình sẽ ngược lại là MUX (2) được phép làm việc cho ra các dữ liệu D_8 hoặc đến D_{15} tùy theo mã $S_2 S_1 S_0$ tương ứng, còn MUX(1) bị cấm. Đầu ra cổng OR ta nhận được $X = Z_{(1)} + Z_{(2)}$ sẽ xuất hiện một trong các dữ liệu D_0 đến D_{15} . Bằng cách tương tự có thể mở rộng cho MUX chọn ví dụ 32 dữ liệu nhờ cách kết hợp 4IC loại này.

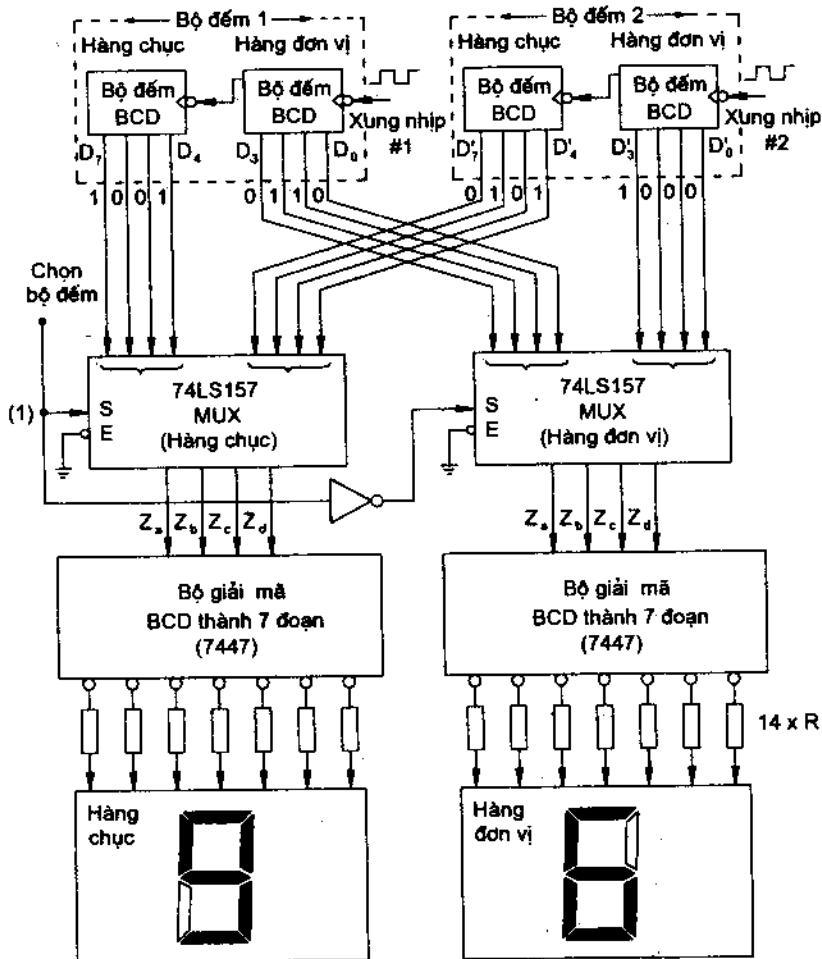
5. Bộ dẫn kênh 2 đầu vào chấp 4 (IC 74157, 74LS157, 74HC157) cho trên hình 3.19a, bên trong có chứa 4 bộ dẫn kênh 2 đầu vào như hình 3.15b đã thể hiện. Ký hiệu logic của IC và bảng chân lý được cho trên hình 3.19b và c.

Ta chú ý cách đặt nhãn dữ liệu cho đầu vào và đầu ra tương ứng của loại IC 74157 đã xét ở trên: khi đó D_0 thể hiện 1 dữ liệu BCD 4 bit (D_{0a}, D_{0b}, D_{0c} và D_{0d}) D_1 thể hiện tương tự 4 bit dữ liệu ($D_{1a} \dots D_{1d}$) tại các đầu vào. Khi tín hiệu ở đầu vào đã chọn địa chỉ $\bar{E}S = 00$. Dữ liệu D_0 được truyền tới đầu ra, còn khi $\bar{E}S = 01$ dữ liệu BCD D_1 được truyền tới các đầu ra. Mức $\bar{E} = 0$ thể hiện sự cho phép (còn $\bar{E} = 1$ thì MUX bị cấm), mức $S = 0$ khi đã có $\bar{E} = 0$ là cho phép D_0 BCD truyền qua MUX, còn khi $S = 1$ (đã có $\bar{E} = 0$) là cho phép D_1 truyền qua MUX. Đây là 1 trường hợp rất có ích trong các ứng dụng của MUX sau này.

3.4.3. Một số ứng dụng quan trọng của MUX

1. Định tuyến dữ liệu: Có nhiều dữ liệu được định hướng tới 1 đích duy nhất, khi đó sử dụng bộ dẫn kênh sẽ cho phép chọn dữ liệu nào (định tuyến đầu vào) hướng tới đích, các dữ liệu không được chọn sẽ bị ngăn lại (cấm) không hoặc chưa tới được đích. Ví dụ : Có hai luồng dữ liệu ở dạng mã BCD 8 bit thể hiện kết quả đếm số xung tại các đầu ra song song của 2 bộ đếm xung, cần chọn một trong hai luồng BCD 8421 này tới đích là một khối giải mã BCD thành 7 vạch và hiển thị kết quả trên 2 màn hình LED tương ứng

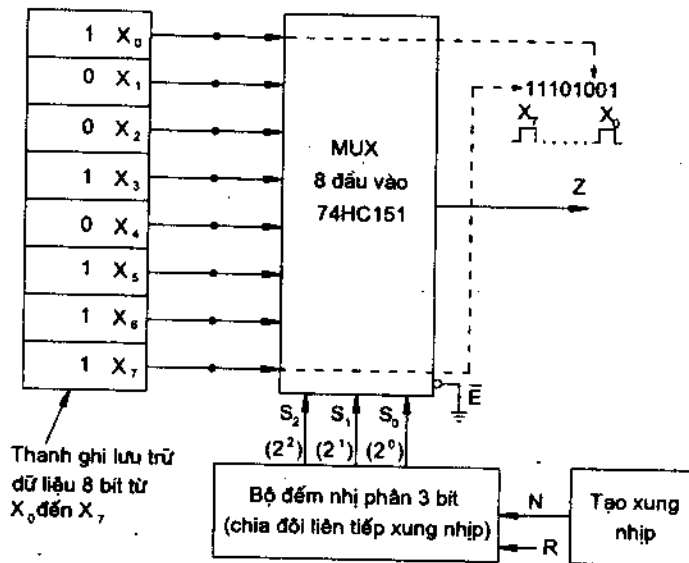
với số thập phân hàng chục và hàng đơn vị. Hình 3.20 là mạch điện cho phép bộ hiển thị sẽ báo kết quả của bộ đếm 1 hay của bộ đếm 2 tùy theo mức của tín hiệu chọn bộ đếm S : khi $S = 1$ dữ liệu D_0 đến D_7 của bộ đếm 1 được chọn đưa tới giải mã và hiển thị kết quả, còn khi $S = 0$ dữ liệu D'_0 đến D'_7 của bộ đếm 2 sẽ được chọn tương tự kết quả trên màn hình lúc này là của bộ đếm 2.



Hình 3.20. Sử dụng MUX 74LS157 để định tuyến cho số liệu đầu ra của hai bộ đếm BCD – 8421 tới giải mã và hiển thị

Ví dụ: Các cổng ra của bộ đếm 1 tại thời điểm đang xét có dãy $D_7 \dots D_0 = 1001\ 0110$, còn tại các cổng ra của bộ đếm 2 lúc này có dãy $D'_7 \dots D'_0 = 0101\ 1000$; khi đó nếu $S = 1$ màn hình LED sẽ hiển thị kết quả số thập phân 96 của bộ đếm 1. Còn nếu $S = 0$ sẽ có kết quả 58 của bộ đếm 2 trên màn hình LED, kết quả ví dụ trên hình vẽ thể hiện khi $S = 1$ và trạng thái của đầu ra đếm như đã cho. Qua ví dụ trên, rõ ràng việc dùng MUX để phân chia thời gian tới đích của các luồng dữ liệu sẽ giúp người thiết kế giảm được các chi phí về linh kiện, tăng độ tin cậy khi vận hành và giảm mức tiêu hao năng lượng một cách đáng kể.

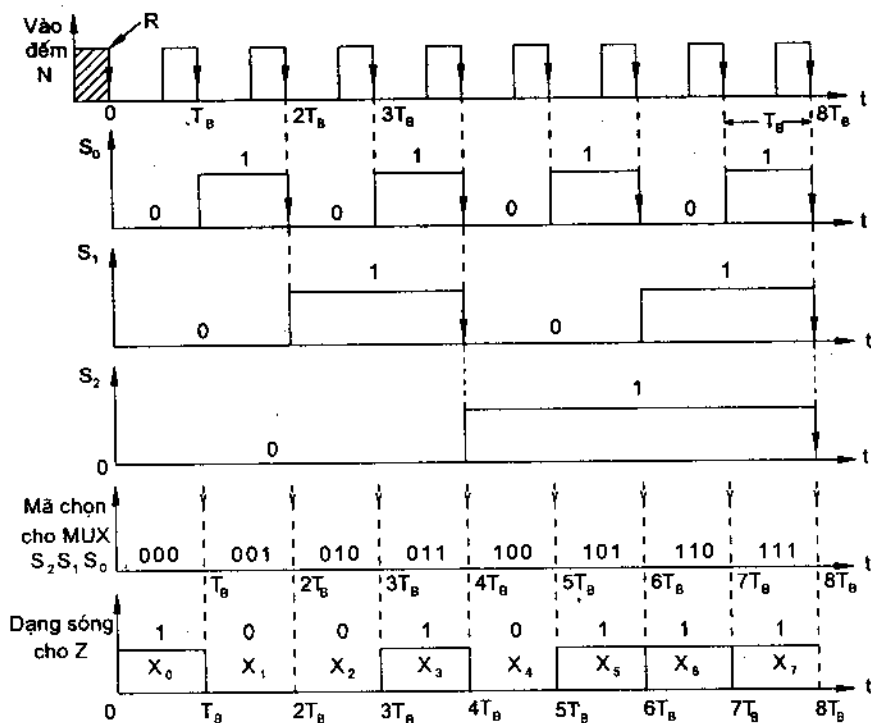
2. Chuyển đổi một luồng dữ liệu từ song song (xuất hiện đồng thời) thành nối tiếp (xuất hiện tuần tự) sử dụng MUX. Một luồng dữ liệu song song có tính chất mọi bit của nó xuất hiện đồng thời, luôn có ưu thế về tốc độ xử lý nhanh nhưng khi truyền nó đi trên một khoảng cách xa sẽ tốn nhiều đường truyền nên thông thường nó được chuyển đổi thành dữ liệu kiểu nối tiếp (với tính chất các bit xuất hiện tuần tự, thời gian giữa 2 bit liên tiếp của dãy gọi là 1 chu kỳ bit) trước khi đưa tới đường truyền. Hình 3.21 cho phép thực hiện biến đổi 8 bit dữ liệu song song thành 1 dãy nối tiếp theo trật tự xác định nhờ các tín hiệu chọn $S_2\ S_1\ S_0$ luân chuyển tuần tự từ trạng thái 000 đến trạng thái 111 theo một chu kỳ xác định nhờ khối tạo xung nhịp (clock) tạo ra xung có chu kỳ là T_B .



Hình 3.21. Chuyển dữ liệu song song 8 bit X_7 đến X_0 thành dãy nối tiếp 8 bit sau 8 nhịp

$$Z = X_7 X_6 X_5 X_4 X_3 X_2 X_1 X_0$$

Đồ thị thời gian minh họa hoạt động của bộ đếm thể hiện trên hình 3.22, đầu ra bộ đếm có trạng thái tuần tự của $S_2 S_1 S_0$ với 8 tổ hợp xuất hiện liên tiếp theo thời gian (với chu kỳ là T_B từ 000 khi $0 \leq t < T_B$ đến 111 khi $7T_B \leq t < 8T_B$) tạo mã chọn cho MUX và do đó tại đầu ra Z sẽ xuất hiện lần lượt các xung $X_0 = 1, X_1 = 0, \dots, X_6 = 1$ và $X_7 = 1$ như thể hiện trên đồ thị hình 3.22 (cần lưu ý xung R xuất hiện đầu tiên trước dãy xung X_i để xóa bộ đếm về trạng thái ban đầu – xem thêm tiết 4.7 chương 4 về bộ đếm).



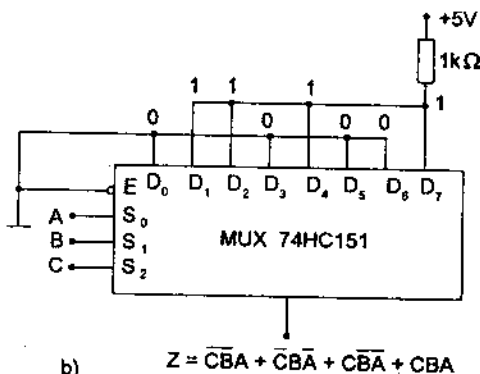
Hình 3.22

3. Sử dụng MUX tạo hàm logic trực tiếp từ bảng chân lý không cần rút gọn, với mục đích này các đầu vào chọn được coi là các biến logic, mỗi đầu vào dữ liệu được nối thường xuyên tới mức cao (1) hay mức thấp (0) tùy theo bảng chân lý. Để làm rõ xét ví dụ sau: Ta muốn có hàm Z : 3 biến A, B, C có bảng chân lý hình 3.23a hàm Z có 4 số hạng bằng 0 là D_0, D_3, D_5, D_6 ứng với các tổ hợp $\bar{C}\bar{B}\bar{A}, \bar{C}BA, C\bar{B}\bar{A}$ và CBA ; như vậy 4 cổng vào dữ liệu tương ứng được nối thường xuyên tới 0V. Các số hạng còn lại có giá trị 1 là $\bar{C}\bar{B}A, \bar{C}B\bar{A}, C\bar{B}A$ và CBA ứng với các cổng vào dữ liệu là D_1, D_2, D_4 và D_7 , ta nối

thường xuyên tới mức nguồn $+V_{cc} = 5V$ (qua điện trở $1k\Omega$ hạn chế dòng), như vậy chỉ các mã địa chỉ ứng với 4 tổ hợp vừa nêu là cho phép giá trị mức 1 đạt được ở đầu ra và bằng cách này đã tạo ra hàm Z muốn có.

Đầu vào chọn			Đầu ra
$C_{(s_2)}$	$B_{(s_1)}$	$A_{(s_0)}$	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

a)



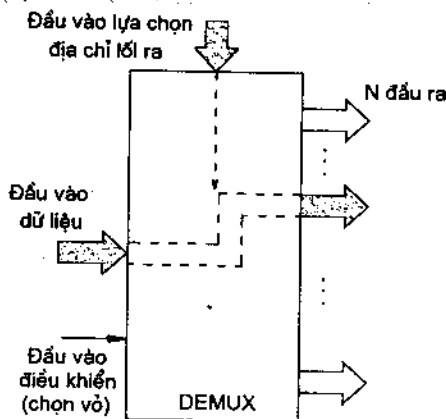
b)

Hình 3.23. a) Bảng chân lý hàm logic muốn có ; b) Thực hiện cùng MUX 74HC151

3.5. BỘ PHÂN PHỐI DỮ LIỆU

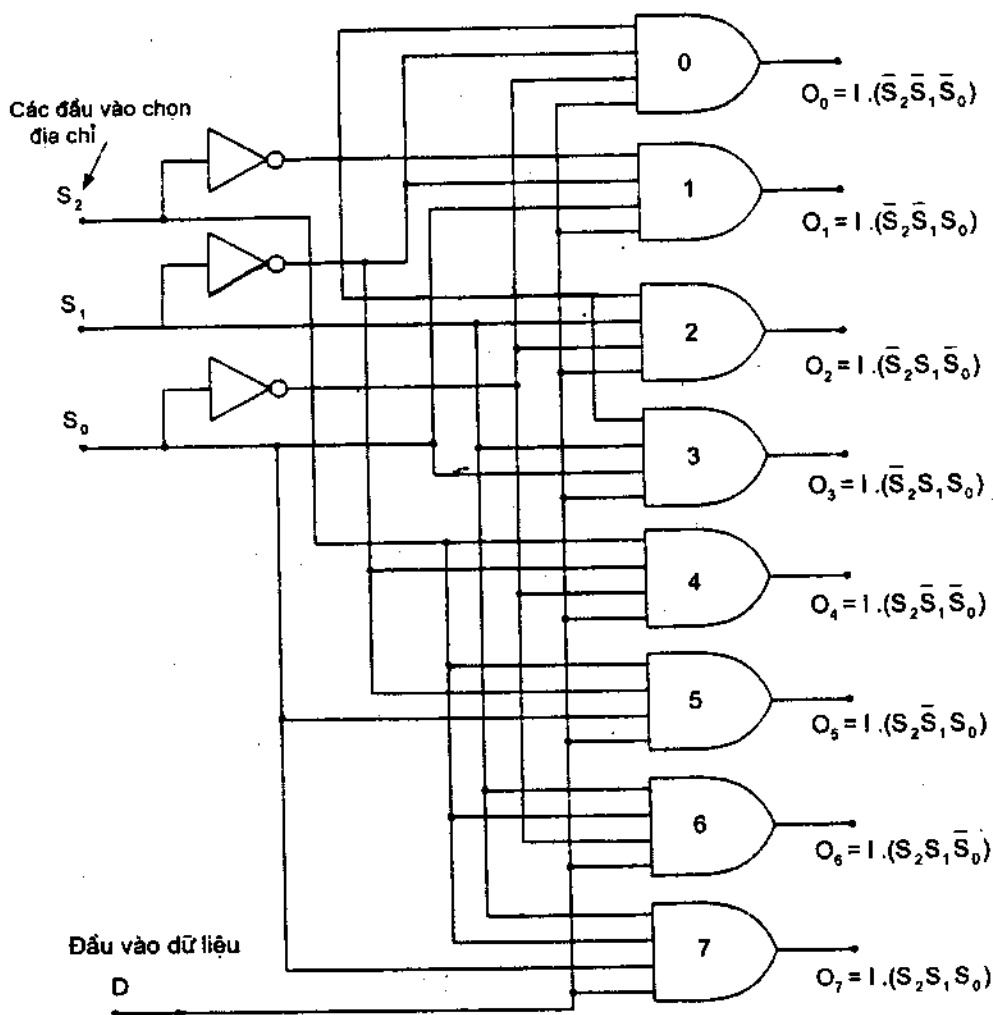
(Phân kênh – demultiplexer – DEMUX)

3.5.1. Bộ DEMUX hoạt động ngược lại với MUX tại mỗi thời điểm chỉ tiếp nhận một đầu vào dữ liệu và theo tín hiệu chọn địa chỉ sẽ phân phối dữ liệu này tới một trong nhiều đầu ra của nó. Như vậy mô hình hoạt động của DEMUX có thể được mô tả ở lược đồ tổng quát hình 3.24, giống như 1 chuyển mạch có nhiều tiếp điểm, tín hiệu vào tại cổng vào dữ liệu chỉ được phép phân phối (truyền) tới 1 trong số N đầu ra do mã địa chỉ đưa tới đầu vào chọn lựa quyết định đầu ra nào được chọn. Với N đầu ra ta sẽ cần n bit để tạo mã địa chỉ cho đầu ra với $2^n \geq N$ trong các bộ phân phối dữ liệu số.



Hình 3.24. Mô tả hoạt động tổng quát của DEMUX

3.5.2. Bộ phân kênh 1 đường thành 8 đường (hình 3.25). Mạch điện sử dụng 8 cổng AND 4 đầu vào, trong đó có 1 đầu vào được tiếp nhận dữ liệu muốn phân phối nhờ DEMUX, các đầu vào còn lại là các đầu vào chọn mã địa chỉ $S_2S_1S_0$ cho việc chọn một trong các đầu ra được tạo ra từ 8 tổ hợp mã của chúng. Các mã địa chỉ của các cổng AND lần lượt là $S_2S_1S_0 = 000$ đến $S_2S_1S_0 = 111$. Mỗi cổng AND có 1 đầu ra được dán nhãn địa chỉ theo $S_2S_1S_0$



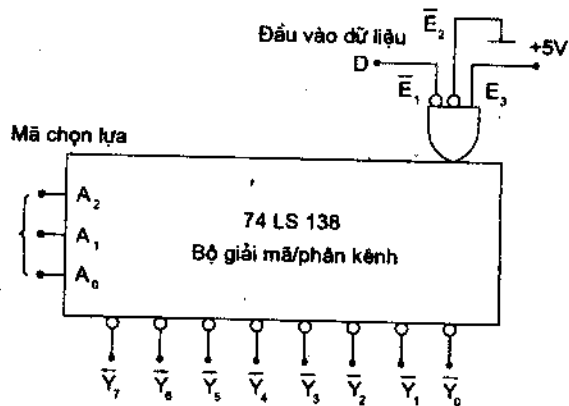
Hình 3.25. Bộ phân kênh 1 đường thành 8 đường

tương ứng, khi các đầu vào của cổng AND nào đó thỏa mãn đủ điều kiện cả 4 đầu vào đều ở mức tích cực cao thì dữ liệu sẽ truyền qua nó tới đầu ra tương ứng của cổng này và với một tổ hợp mã chọn $S_2S_1S_0$ xác định, chỉ có duy nhất 1 cổng AND thỏa mãn điều kiện trên.

3.5.3. Sử dụng bộ giải mã 1 trong 8

Có thể sử dụng bộ giải mã 1 trong 8: IC 74LS138 thực hiện nhiệm vụ của DEMUX như mạch điện hình 3.25 đã phân tích. Như thấy trên hình 3.26 ứng với 1 trong các tổ hợp của mã chọn $A_2A_1A_0$, đầu ra Y_i ($i = \overline{0,7}$) sẽ xuất hiện mức tích cực 0. Tín hiệu được đưa tới đầu vào $\overline{E}_1$ đóng vai trò là đầu vào dữ liệu (ở đây bộ điều kiện $\overline{E}_2 = 0$ và $E_3 = 1$ đã luôn luôn được bảo đảm ở trạng thái tích cực, chỉ còn lại $\overline{E}_1$ phụ thuộc vào mức dữ liệu là cao hay thấp). Khi xuất hiện tổ hợp mã chọn (ví dụ $A_2A_1A_0 = 100$ tại các đầu vào chọn lựa) lúc đó chỉ duy nhất đầu ra $\overline{Y}_4$ ở mức tích cực thấp còn lại các đầu ra khác đều ở mức 1 (không được kích hoạt). Dữ liệu tác động vào $\overline{E}_1$ lúc này nếu ở mức thấp $\overline{Y}_4$ cũng sẽ ở mức thấp, nếu $\overline{E}_1$ ở mức cao thì $\overline{Y}_4$ sẽ lên mức cao. Nghĩa là đầu ra $\overline{Y}_4$ làm việc theo (nhịp) cao hay thấp của tín hiệu dữ liệu, chúng hoàn toàn tương đương nhau: $\overline{Y}_4 = D$. Các đầu ra khác $\overline{Y}_4$ luôn luôn ở mức cao nên không làm việc (không được kích hoạt). Ta có thể phân tích tương tự cho các mã $A_2A_1A_0$ còn lại.

Nếu tại đầu vào dữ liệu $D = \overline{E}_1$ trong mạch hình 3.26 được đặt tới đây xung vuông góc có chu kỳ T_0 ổn định (xung bất nhịp) thì mỗi khi có một mã địa chỉ $A_2A_1A_0$ áp tới đầu vào chọn, tại cổng ra $\overline{Y}_i$ ($i = \overline{0,7}$) tương ứng với địa chỉ được chọn sẽ xuất hiện xung nhịp T_0 . Mạch thực hiện nhiệm vụ phân phối xung nhịp T_0 tới 1 trong 8 tuyến khác nhau nhằm đồng bộ cho các thiết bị khác.



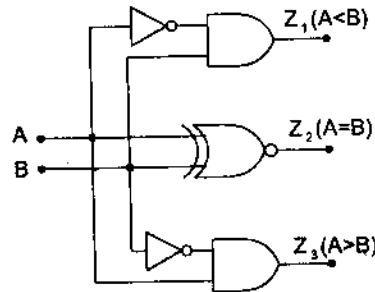
Hình 3.26. Sử dụng 74LS138 làm nhiệm vụ phân kênh

3.6. BỘ SO SÁNH HAI SỐ NHỊ PHÂN CÙNG CẤP (không dấu)

3.6.1. Bài toán so sánh 2 số chỉ đặt ra với 2 số cùng cấp. Thuật toán so sánh 2 số nhị phân cùng cấp giống như so sánh 2 số thập phân cùng cấp. Thoạt tiên so chúng ở cấp cao nhất, nếu bằng nhau thì so sánh tiếp ở cấp thấp hơn kế nó, cho đến cấp thứ k nếu chúng khác nhau thì đó là kết quả chung của phép so sánh. Tóm lại việc so sánh 2 số nhị phân cần quay về so sánh 2 bit nhị phân đồng cấp (cùng trọng số). Bảng chân lý bộ so sánh 2 bit nhị phân cho trên hình 3.27a và mạch logic thực hiện được thể hiện ở hình 3.27b.

Biến vào		Hàm ra		
A	B	$Z_3(A>B)$	$Z_2(A=B)$	$Z_1(A<B)$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

a)



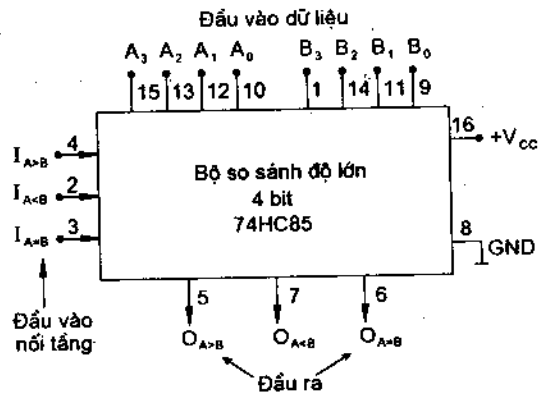
b)

Hình 3.27. Bộ so sánh 2 bit

a) Bảng chân lý bộ so sánh 2 bit nhị phân ; b) Mạch điện bộ so sánh 2 bit nhị phân.

Việc so sánh tiến hành theo thủ tục đã nói trên gọi là phương pháp so sánh nối tiếp thực hiện từng bit một và do đó tốc độ hoạt động chậm với các số nhiều bit.

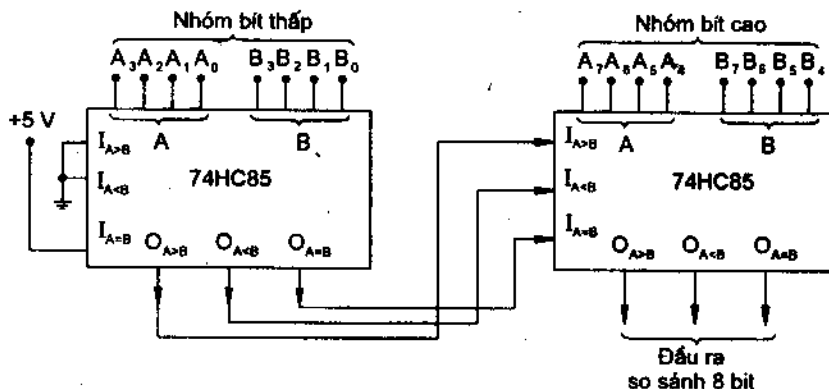
3.6.2. Để tăng tốc độ cần sử dụng phương pháp so sánh đồng thời hoặc kết hợp so sánh đồng thời từng nhóm bit nhỏ (4 bit) sau đó so nối tiếp các nhóm từ cao tới thấp sẽ hợp lý hơn. Nhóm IC MSI : 7485, 74LS85 hay 74HC85 được chế tạo nhằm giải quyết nhiệm vụ so sánh các số nhị phân không dấu có 4 bit. Ký hiệu logic của 74HC85 cho trên hình 3.28 và bảng chân lý của IC được cho trong bảng 3.7.



Hình 3.28. Ký hiệu logic của IC so sánh độ lớn 4 bit 7485, 74LS85, 74HC85.

- 8 dữ liệu đầu vào là 2 số nhị phân 4 bit.

$A = A_3 A_2 A_1 A_0$ và $B = B_3 B_2 B_1 B_0$ được đưa đồng thời tới các đầu vào dữ liệu tương ứng.



Hình 3.30. Nối liên tiếp 2 IC 74HC85 tạo bộ so sánh 8 bit

Khi ở bộ so sánh các bit cao đã có kết quả, theo thuật toán so sánh đó chính là kết quả chung $A < B$ mà không cần xem xét tới các nhóm bit thấp hơn kết quả so sánh ra sao nữa.

Ví dụ 2: Với $A = 1010\ 1110$

$B = 1010\ 1001$

Bộ so sánh nhóm các bit cao của A và B sẽ có kết quả như nhau ($A_7A_6A_5A_4 = B_7B_6B_5B_4$) khi đó cần xem xét kết quả so sánh của nhóm 4 bit thấp: $A_3A_2A_1A_0 = 1110$ và $B_3B_2B_1B_0 = 1001$, tại đầu ra

$O_{A>B} = 1 = I_{A>B}$. Tại đầu vào nối tiếp của bộ so sánh bit cao, do đầu vào dữ liệu của chúng bằng nhau nên theo bảng chân lý 3.7 xuất hiện mức cao tại đầu ra chung cho 8 bit $O_{A>B} = 1$ thông báo kết quả phép so sánh là $A > B$.

3.6.4. Ví dụ ứng dụng của bộ so sánh

Ví dụ 1 : Trong máy photocopy, để điều khiển số lượng bản chụp, mạch điện hình 3.31 được sử dụng : Ban đầu người chụp đặt trước số bản sẽ chụp nhờ hệ thống phím bấm ký tự thập phân – qua bộ mã hoá ký tự này được chuyển đổi thành mã BCD 8421 đưa tới nhóm đầu vào A của bộ so sánh (nếu cần con số thập phân hàng chục nhóm A sẽ có 8 bit, hàng trăm có 12 bit và hàng nghìn có 16 bit biểu diễn). Đầu vào B của bộ so sánh được lấy ra từ bộ đếm xung chụp (đếm BCD) tương ứng với số lượng bản đã chụp. Tại đầu ra của bộ so sánh, tín hiệu có hai khả năng: nếu $A > B$ thì logic điều khiển vẫn cho mức 1 điều khiển máy hoạt động tiếp tục chụp, còn nếu khi $A = B$ thì logic điều khiển phát tín hiệu tắt bộ phận chụp không hoạt động, tín hiệu xóa xuất hiện đưa bộ đếm BCD về trạng thái ban đầu, chờ 1 nhịp đặt phím làm việc mới hoặc chờ nhấn tiếp nút

Bảng 3.7. BẢNG CHÂN LÝ CỦA IC SO SÁNH ĐỘ LỚN 4 BIT 74HC 85

Đầu vào so sánh				Đầu vào nối tiếp tầng			Đầu ra		
A_3B_3	A_2B_2	A_1B_1	A_0B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$O_{A>B}$	$O_{A<B}$	$O_{A=B}$
$A_3 > B_3$	x	x	x	x	x	x	1	0	0
$A_3 < B_3$	x	x	x	x	x	x	0	1	0
$A_3 = B_3$	$A_2 > B_2$	x	x	x	x	x	1	0	0
$A_3 = B_3$	$A_2 < B_2$	x	x	x	x	x	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	x	x	x	x	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	x	x	x	x	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	x	x	x	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	x	x	x	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	0	1	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	0	0	0	0

- IC 74HC85 có 3 đầu ra tích cực mức cao thể hiện 3 kết quả của phép so sánh (ví dụ khi $O_{A>B} = 1$ kết quả so sánh chung là $A > B$).

- 3 đầu vào nối tầng để mở rộng phép so sánh ra với số bit nhiều hơn 4 khi nối tiếp 2 bộ so sánh 4 bit với nhau. Các đầu vào nối tầng có ký hiệu nhãn trùng với nhãn của các đầu ra ($A > B$, $A < B$ và $A = B$).

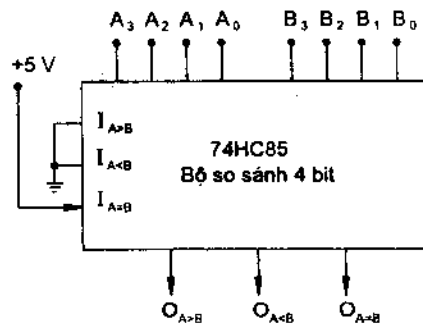
3.6.3. Cách nối IC 74HC85 thực hiện so sánh các số 4 bit và 8 bit

Ví dụ 1:

Với $A = 10101111 = A_7A_6A_5A_4A_3A_2A_1A_0$

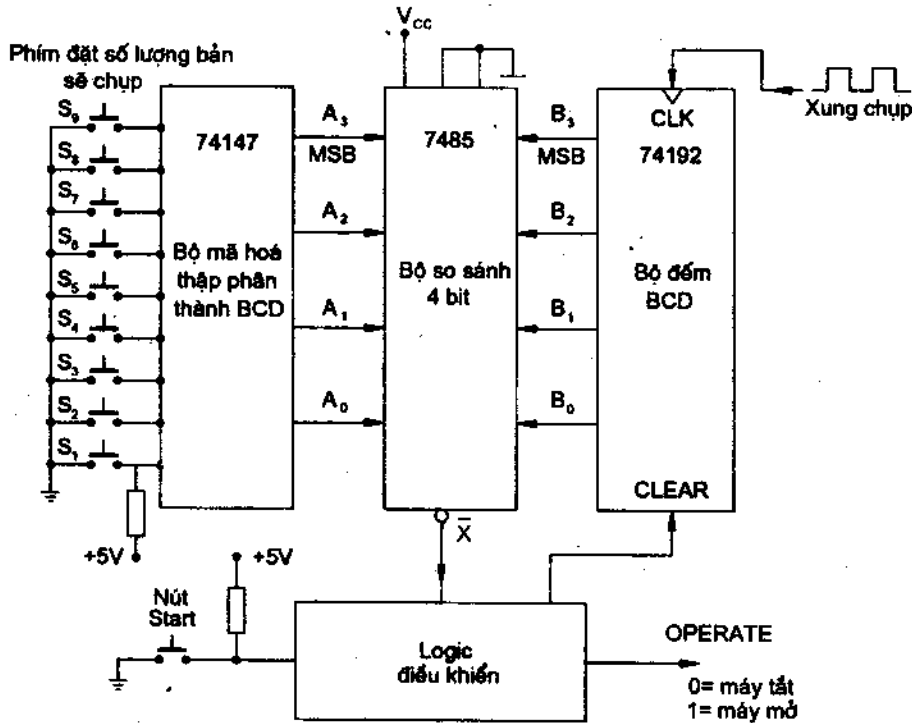
$B = 10110001 = B_7B_6B_5B_4B_3B_2B_1B_0$

Hoạt động của mạch 3.30 như sau: Bộ so sánh bit cao thực hiện so sánh 4 bit già của A và B là A_7B_7 , A_6B_6 , A_5B_5 , A_4B_4 , tại đầu ra $O_{A<B} = 1$, lúc đó tất cả các mức logic từ đầu vào nối tầng đưa kết quả từ việc so sánh các bit thấp (trường hợp này là $O_{A>B} = 1$) sẽ bị cấm vì kết quả đã được xác định ở nhóm bit cao là $O_{A<B} = 1$.



Hình 3.29. Cách nối của IC 74HC85 thực hiện nhiệm vụ so sánh 4 bit song song.

khởi động (chụp lại số bản đã chỉ định trước). Trên thực tế máy có thể đặt trước số lượng hàng trăm hay hàng chục bản khi đó bộ so sánh sẽ được mở rộng bằng cách ghép nhiều bộ so sánh 4 bit lại.



Hình 3.31. Mạch điện điều khiển số tượng bản chụp của máy photocopy

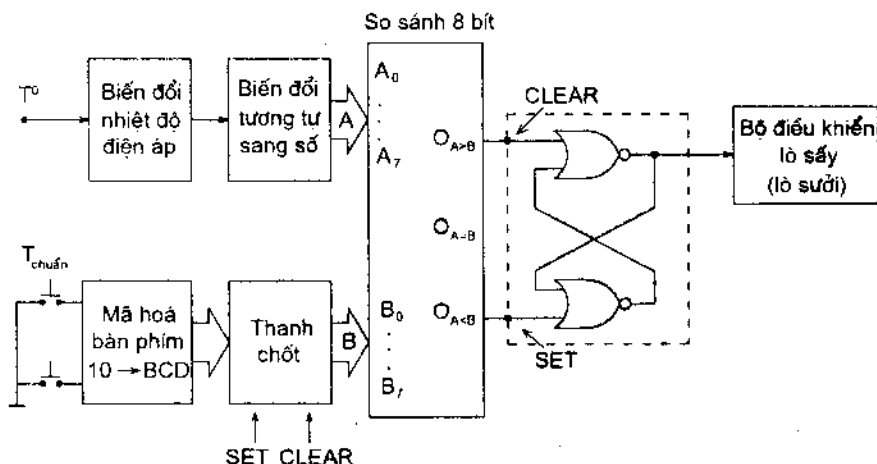
Ví dụ 2: Điều khiển nhiệt độ dùng bộ so sánh số. Mạch điện cấu trúc khối hệ điều khiển nhiệt độ cho trên hình 3.32.

Nhóm bit dữ liệu A áp vào bộ so sánh mang thông tin thực về nhiệt độ của đối tượng T_X sau khi đã thực hiện 2 quá trình biến đổi :

- Biến đổi từ nhiệt độ sang mức điện thế tỷ lệ (ở bộ cảm biến nhiệt độ).
- Biến đổi từ điện thế thành số nhị phân BCD biểu diễn giá trị điện thế (ở bộ biến đổi tương tự – số).

Nhóm bit dữ liệu B áp vào bộ so sánh mang thông tin về một mức nhiệt độ $T_{\text{chuẩn}}$ đặt trước (dự kiến) cho hệ theo dõi T_X đạt đến $T_{\text{chuẩn}}$, không được phép vượt quá. Tái của quá trình có thể là một bộ điều khiển lò sấy (ví dụ cuộn nung) nếu T_X là nhiệt độ sấy một sản phẩm nào đó, hoặc một cơ cấu thực hiện tác động trực tiếp tới T_X .

Khi $A < B$ mức tích cực cổng ra bộ so sánh $O_{A<B} = 1$ thông báo việc $T_X < T_{\text{chuẩn}}$ và bộ điều khiển tải tiếp tục vận hành cho phép T_X tăng. Khi $A \geq B$ mức $O_{A<B} = 1$ tạo tín hiệu CLEAR xóa thanh chốt dữ liệu $B_7 - B_0$, bộ điều khiển tải ngừng hoạt động, cho tới khi T_X giảm tương ứng với lúc $A < B$ sẽ vận hành lại.



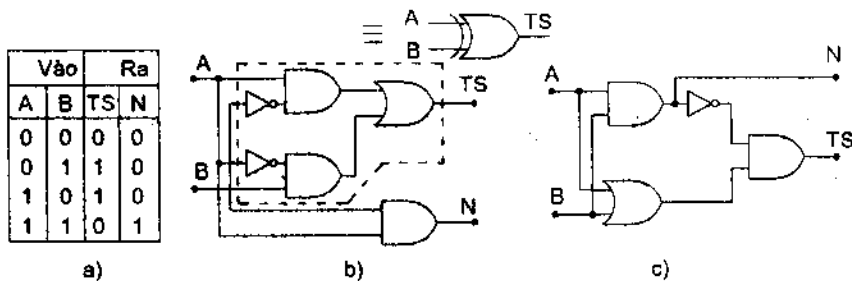
Hình 3.32. Hệ thống điều khiển số nhiệt độ dùng bộ so sánh 8 bit

3.7. CÁC BỘ CỘNG SỐ HỌC NHỊ PHÂN

3.7.1. Mạch bán cộng (bán tổng)

Mạch cộng số học là mạch điện cơ bản để từ đó có thể thực hiện các phép toán số học khác trong số học nhị phân. Mạch bán cộng thực hiện cộng hai số nhị phân tự nhiên 1 bit, có bảng chân lý và mạch điện thực hiện cho trên hình 3.33 gồm 2 đầu vào số liệu là bit A và bit B, hai đầu ra cho tổng số (TS) và số nhớ (N). Từ kết quả ở chương 1, 2, ta có hàm ra:

$$\begin{aligned}
 TS &= A \cdot \bar{B} + \bar{A} \cdot B = A \cdot \bar{B} + B \cdot \bar{A} + \bar{A}B + A\bar{A} \\
 &= \bar{B}(A + B) + \bar{A}(A + B) = (\bar{A} + \bar{B})(A + B) \\
 TS &= \overline{A \cdot B}(A + B); N = A \cdot B
 \end{aligned} \tag{3.7}$$



Hình 3.33. Bảng chân lý và mạch điện bộ bán tổng
a) Bảng chân lý bộ bán tổng ; b) Mạch điện bộ bán tổng (cách 1) ;
c) Mạch điện bộ bán tổng (cách 2 theo hệ thức 3.7).

3.7.2. Mạch cộng đầy đủ (gọi tắt là FA) sử dụng khi phải cộng hai số nhị phân nhiều bit. Mạch cộng đầy đủ có bảng chân lý cho trên hình 3.34a với mạch điện thực hiện bảng cho trên hình 3.34b và c. Bộ cộng đầy đủ gồm 3 đầu vào dữ liệu: 2 đầu dành cho 2 biến nhị phân cùng trọng số cần cộng, đầu vào thứ 3 là tín hiệu truyền nhớ từ cấp nhị phân thấp hơn liền kề gửi tới. Mạch có hai đầu ra thể hiện tổng số ở nhịp đang cộng và đầu ra tạo tín hiệu nhớ (nếu có) gửi tới nhịp tương lai sắp cộng.

Hệ hàm ra của bộ cộng đầy đủ có dạng:

$$\begin{aligned}
 TS_i &= \bar{A}_i \bar{B}_i N_{i-1} + \bar{A}_i B_i \bar{N}_{i-1} + A_i \bar{B}_i \bar{N}_{i-1} + A_i B_i N_{i-1} \\
 &= (A_i \oplus B_i) \oplus N_{i-1}
 \end{aligned} \quad (3.8)$$

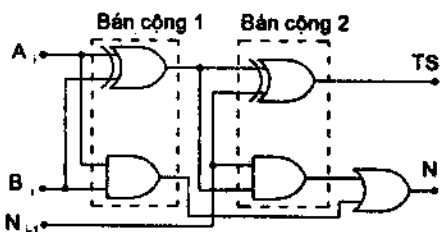
$$\begin{aligned}
 N_i &= A_i B_i \bar{N}_{i-1} + \bar{A}_i B_i N_{i-1} + A_i \bar{B}_i N_{i-1} + A_i B_i N_{i-1} \\
 &= (A_i \oplus B_i) \cdot N_{i-1} + A_i B_i
 \end{aligned}$$

Ở đây chỉ số i thể hiện nhịp đang cộng, $i-1$ là nhịp trước đó (cấp nhị phân đã được cộng có trọng số thấp hơn 1 bậc nhị phân) và $i+1$ thể hiện tương lai (nhịp cộng 2 bit có trọng số cao hơn hiện tại 1 cấp).

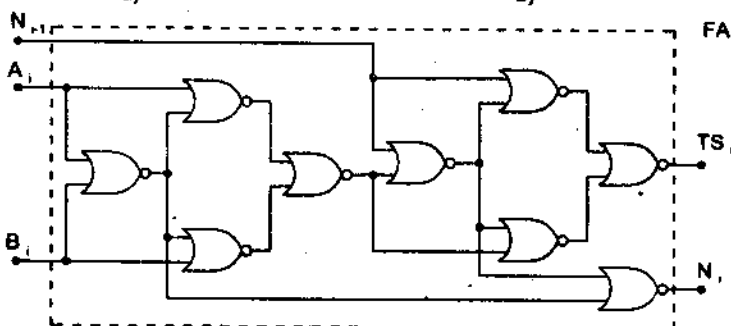
Từ hệ hàm ra (3.8) ta xây dựng được mạch điện hình 3.34b bao gồm 2 bộ bán tổng và 1 cổng OR. Hình 3.34c là bộ cộng đầy đủ thực hiện chỉ từ các cổng NOR.

Vào			Ra	
A _i	B _i	N _{i-1}	TS _i	N _i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

a)



b)



c)

Hình 3.34. Bộ cộng đầy đủ

a) Bảng chân lý bộ cộng đầy đủ ; b) Mạch điện bộ cộng đầy đủ xây dựng từ hai bộ bán cộng bổ sung 1 cổng OR ; c) Bộ cộng đầy đủ xây dựng từ các cổng NOR mang tính chất đồng nhất và kinh tế cao, công nghệ thực hiện tối ưu.

3.7.3. Thực hiện phép toán cộng, trừ

1. Khi thực hiện cộng 2 số nhị phân tự nhiên A và B có trường hợp xảy ra : thường phép cộng quy định được phép với 1 số lượng bit hữu hạn (ví dụ 4 bit) nếu tổng $A + B$ nhỏ hơn hay bằng giới hạn tối đa là $2^4 - 1$ thì phép cộng là không tràn, khi $A + B > 2^4 - 1$ (với 4 bit) sẽ xảy ra hiện tượng tràn (kết quả phép cộng được biểu diễn bởi 1 số 5 bit) khi đó cần phải dò tìm và thông báo sửa chữa nếu không kết quả sẽ sai.

2. Với 2 số nhị phân A và B có 4 trường hợp thực hiện phép cộng là :

a) A và B đều là số dương (thể hiện bit đầu tiên giá trị 0)

Tiến hành cộng bình thường:

↓ bit dấu

- Không tràn : $A = +0011 = 0\ 0011$

$B = +0100 = 0\ 0100$

$A + B = +0111 = 0\ 0111$

- Bị tràn $A = +1001 = 0\ 1001$

$B = +1000 = 0\ 1000$

$A + B = +0001 = 1\ 0001$

↓ Theo quy ước 1 chỉ số âm, kết quả là sai (-0001)

Khi đó cần thông báo có tràn (Over Flow) để sửa chữa kết quả.

b) A dương, B âm và $A > B$ kết quả tổng sẽ là một số dương và không có tràn. Lúc đó chuyển B về số bù 1 hay bù 2 sau đó thực hiện như trường hợp trên với 2 số dương (khi tràn).

c) A dương, B âm và $A < B$ kết quả tổng là 1 số âm và không bị tràn, cần chuyển B về dạng mã bù 1 hay bù 2 để thay bằng cộng hai số không dấu (cùng dương).

d) A âm, B âm kết quả là 1 số âm, có thể không hoặc có tràn, cần có mạch dò tìm trạng thái tràn để thông báo xử lý kết quả.

Khi thực hiện các quy tắc cộng cần lưu ý hai quy tắc : số dương sẽ có thêm 1 bit dấu trị 0 ở trước số đã cho, số âm có bit dấu trị 1.

Số bù 1 là số nhị phân sau khi đổi trị mọi bit của nó.

Số bù 2 là số bù 1 sau đó cộng thêm 1 (xem chương 1).

Trường hợp trừ 2 số nhị phân thực chất là trường hợp (b) và (c) nêu trên vì $A - B = A + (-B)$

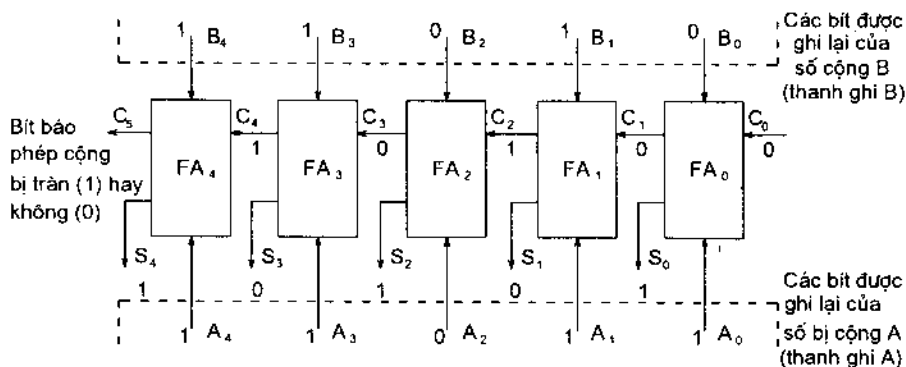
Vấn đề số học nhị phân được xét kỹ trong các giáo trình liên quan về bộ xử lý số học và cấu trúc máy tính số.

3.7.4. Sử dụng bộ cộng đầy đủ (Full Adder – FA) trong phép cộng nhị phân

Tiến trình tiêu biểu việc cộng nhị phân là: bắt đầu từ việc cộng 2 bit LSB (thấp nhất) theo quy tắc (1.2) ở chương 1, các bit ở cấp tiếp theo cần bổ sung số hạng N_{i-1} của cấp trước nếu có gửi đến, làm tuần tự từ cấp thấp nhất LSB đến cấp nhị phân cao nhất MSB ta có kết quả.

– Nếu thực hiện quy trình như trên ta có phương pháp cộng kiểu nối tiếp – tốc độ chậm vì số nhị phân n bit cần n nhịp cộng mới cho tổng kết quả.

– Việc cộng song song là đồng thời thực hiện với tất cả hay một nhóm nhiều bit một lúc sẽ cho kết quả tổng nhanh hơn nhiều. Cấu hình thực hiện theo phương pháp cộng song song được cho trên hình 3.35.



Hình 3.35. Sơ đồ khối của mạch cộng song song 5 bit dùng bộ cộng FA. Tổng kết quả xuất hiện lại các đầu ra S_4, S_3, S_2, S_1, S_0

(Ghi chú: Thanh ghi sẽ được trình bày ở chương tiếp sau với hệ logic dây)

Trên hình 3.35 ta đã ví dụ:

Tại thanh ghi A có $A = 1\ 1\ 0\ 1\ 1 = A_4\ A_3\ A_2\ A_1\ A_0 = 27_{10}$

Tại thanh ghi B có $B = 1\ 1\ 0\ 1\ 0 = B_4\ B_3\ B_2\ B_1\ B_0 = 26_{10}$

Tín hiệu kết quả tại các cổng ra là :

$$S = S_4\ S_3\ S_2\ S_1\ S_0 = 1\ 0\ 1\ 0\ 1$$

Tín hiệu nhớ tại các cổng vào của các FA là :

$$C_0 = 0 ; C_1 = 0 ; C_2 = 1 ; C_3 = 0 ; C_4 = 1 \text{ và } C_5 = 1 \text{ (bit báo tràn)}$$

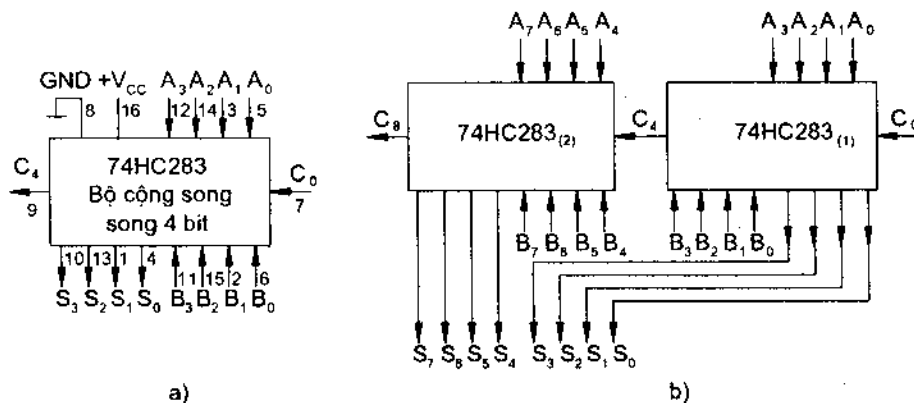
Như vậy tại MSB có tín hiệu nhớ $C_5 = 1$ gửi tới cấp nhị phân tiếp sau ($A_5 + B_5$) nếu có, hoặc $C_5 = S_5$ nếu không thực hiện tiếp (phép cộng chỉ dừng lại với 5 bit). Kết quả chung sẽ là $S = S_5\ S_4\ S_3\ S_2\ S_1\ S_0 = 1\ 1\ 0\ 1\ 0\ 1 = 53_{10}$.

Kết quả S thường được ghi lại bằng 1 thanh ghi kết quả S theo phương thức truyền song song S_i tới các ô nhớ của thanh ghi, đôi khi dữ liệu từ A (số bị cộng) đã được đưa vào bộ cộng, lúc đó thanh ghi A đang trống có thể sử dụng để ghi luôn kết quả S_i vào nó và có vài trò như 1 số hạng (hay thừa số) của một phép cộng (hay nhân) tiếp sau phép cộng đã thực hiện.

– Hiện tượng cần quan tâm và khắc phục trong cấu trúc hình 3.35 là sự trễ tích lũy tín hiệu truyền nhớ C_i khi đi từ FA thấp đến các FA cao và do đó xuất hiện bị muộn hơn so với các số hạng tương ứng của FA ở cấp cao (ví dụ giữa A_3, B_3 áp vào FA_3 sớm so với C_3 từ FA_2 chuyển tới trễ hơn). Người ta khắc phục bằng các mạch logic kiểm tra trước bit nhớ: trả lời xem trong nhóm bit thấp $A_2A_1A_0$ sẽ cộng với $B_2B_1B_0$ liệu có tồn tại bit nhớ C_2 hay không? nếu có thì cho phép bit nhớ này xuất hiện sớm hơn bằng con đường truyền khác với thời gian chậm trễ nhỏ hơn nhiều khi đi theo đường truyền $FA_0 \rightarrow FA_1 \rightarrow FA_2 \rightarrow C_2$, nhờ cải tiến đó, tốc độ của bộ cộng song song được cải thiện nhanh lên rõ rệt.

3.7.5. Nối tăng các bộ cộng song song

Khi cần cộng các số nhị phân nhiều bit, cần dùng phương pháp nối tăng nhiều bộ cộng song song như thể hiện trên hình 3.36.



Hình 3.36

a) IC cộng 4 bit song song 74HC283 ; b) Ghép 2 IC cộng 4 bit thành bộ cộng 8 bit

Trên hình 3.36b bộ cộng dùng IC 74HC283 (1) thực hiện cộng 4 bit trễ. Còn IC(2) thực hiện cộng 4 bit già của 2 số nhị phân 8 bit:

$$A = A_7A_6A_5A_4A_3A_2A_1A_0 \text{ và } B = B_7B_6B_5B_4B_3B_2B_1B_0$$

C_4 là bit nhớ sau phép cộng $(A_3+B_3+C_3)$ được gửi tới phép cộng A_4B_4 kế tiếp.

C_8 là tín hiệu truyền nhớ (báo tràn) tại phép cộng 2 bit MSB của 2 số với C_7 gửi tới.

3.7.6. Dùng bộ cộng IC 74283 thực hiện phép trừ hai số nhị phân

1. Khi thực hiện phép trừ hai số nhị phân 4 bit $A = A_3A_2A_1A_0$ (số bị trừ) và $B = B_3B_2B_1B_0$ (số trừ), trước tiên ta lập số bù 2 của B sau đó thực hiện phép $A - B = A +$ (bù 2 của B) nhờ IC 74HC283.

Hình 3.37a mô tả quá trình cộng hai số nhị phân âm và dương.

Ví dụ $A = (-3)_{10}$ và $B = (+6)_{10}$

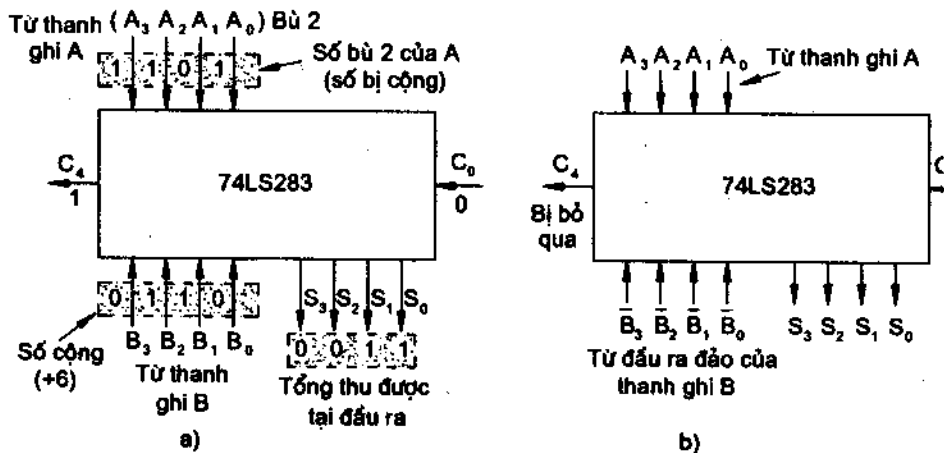
2. Từ hình 3.37a ta thấy các bước tiến hành:

* Đổi các bit nhị phân của $A < 0$ thành mã bù 2 của nó bằng cách lập bù 1 sau đó cộng thêm 1 vào số bù 1, dấu trừ thể hiện ở bit dấu giá trị 1.

$$A = 1\ 0\ 1\ 1 \rightarrow \bar{A} = 1\ 1\ 0\ 0 \rightarrow \bar{A} + 1 = 1\ 1\ 0\ 1$$

* Đưa các ký số nhị phân của $A_{(\text{bù } 2)}$ vào bộ cộng song song đồng thời với B để thực hiện phép cộng $A_{(\text{bù } 2)} + B = 1\ 1\ 0\ 1 + 0\ 1\ 1\ 0$

bit dấu của $B > 0$



Hình 3.37 a) Phép cộng số âm và 1 số dương trong hệ 2 (cộng song song);
b) Phép trừ dùng bộ cộng song song

* Tại đầu ra tổng có kết quả $S = 0\ 0\ 1\ 1$, $C_4 = 1$.

* Bỏ qua $C_4 = 1$ ta có kết quả $S = 0\ 0\ 1\ 1$ là kết quả đúng của phép cộng.

3. Việc thực hiện phép tính $A - B$ được thể hiện trên hình 3.37b với bộ cộng song song 4 bit 74LS283. Trong cấu trúc này, việc lập mã bù 2 của số trừ B thực hiện tự động bằng cách:

* Lấy các giá trị $\bar{B}$ tại các đầu ra đảo của thanh ghi B ta có $\bar{B}_3 \bar{B}_2 \bar{B}_1 \bar{B}_0$

* Cho đầu vào nhớ $C_0 = 1$ (cộng thêm 1 cho $\bar{B}_0$)

* Phép cộng $A + (\bar{B} + 1)$ đồng thời được xử lý trong IC chính là thực hiện phép trừ $(A - B)$ muốn có.

* Bit đầu tiên trong kết quả vẫn thể hiện bit dấu của hiệu số.

3.7.7. Bộ cộng số nhị phân BCD 8421

1. Quá trình cộng 2 số dạng BCD 8421 4 bit thực hiện như sau:

– Sử dụng phép cộng nhị phân thông thường cho mỗi nhóm 4 bit ký tự (của một chữ các mã thể hiện).

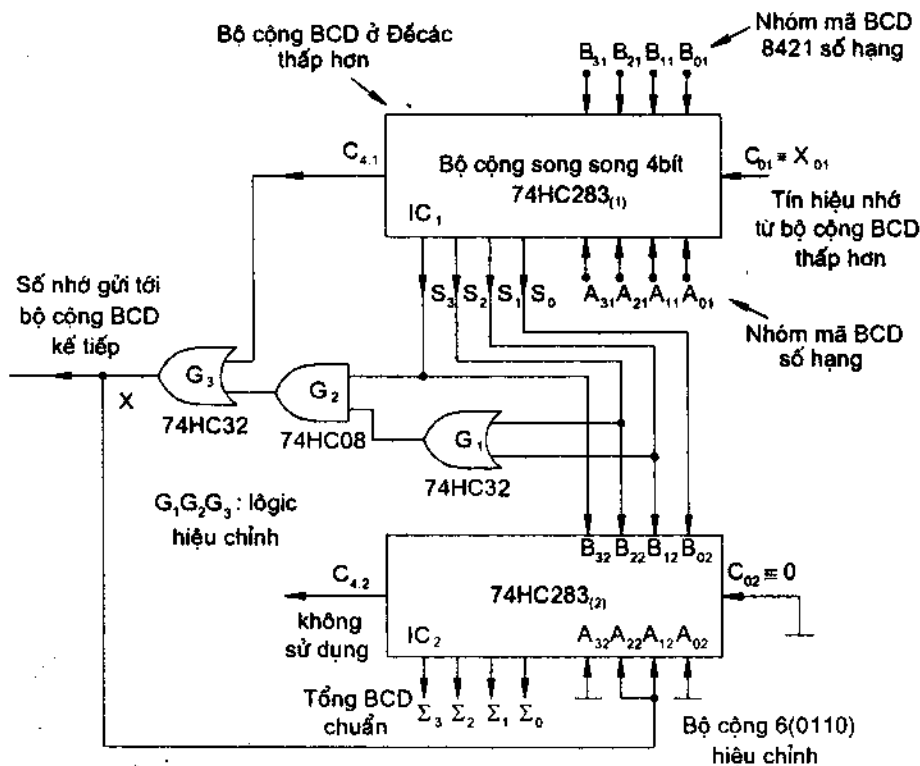
– Ở những nhóm chữ các tương ứng đã cộng nếu tổng thu được ≤ 9 thì kết quả cộng nhị phân thường trong nhóm đó là chuẩn BCD.

– Khi tổng 2 nhóm chữ các cùng cấp là > 9 phải cộng thêm $0110 = 6_{10}$ vào kết quả để cho ra kết quả của BCD chuẩn, tổng mới này sẽ tạo số nhớ sang nhóm chữ các cấp cao hơn kế tiếp.

2. Mạch logic thực hiện: ngoài chức năng cộng nhị phân 8421 thông thường phải có thêm mạch logic bổ sung nhằm phát hiện trạng thái BCD "giả" > 9 để thực hiện phép cộng 6 (0110) bổ sung. Hình (3.38) là mạch điện bộ cộng BCD 4 bit có logic tạo nhớ và hiệu chỉnh từ mã BCD 8421 không chuẩn cho phép cộng nhị phân thông thường gây ra (các từ mã bị cấm không có trong biểu diễn của BCD 8421 là từ $10_{10} = 1010$ đến $15_{10} = 1111$).

– Mạch sử dụng 2 IC cộng 4 bit song song loại 74HC 283, một trong chúng để thực hiện phép cộng nhị phân thông thường, bộ cộng thứ hai sẽ hoạt động thực hiện phép cộng thêm 6 (0110) khi được phép (phát hiện thấy ở bộ cộng trên các từ mã bị cấm từ 1010 đến 1111 tại cổng $S_3S_2S_1S_0$ của nó).

Giả sử bộ cộng đang thực hiện cộng 2 nhóm mã BCD 4 bit ở 1 để các nào đó có giá trị (lấy từ thanh ghi A và B) là $A_3A_2A_1A_0$ và $B_3B_2B_1B_0$ (với cả tín hiệu nhớ C_0 từ một phép cộng 2 nhóm cấp thấp hơn mười lần), kết quả tại đầu ra của IC₁ có tổng $S_4S_3S_2S_1S_0$. Giá trị của tổng $S_4S_3S_2S_1S_0$ chỉ có thể biến thiên trong khoảng 0000 đến $1001 + 1001 = 10010$ (do tính chất của nhóm mã BCD 8421).



Hình 3.38. Bộ cộng BCD 8421 chứa mạch phát hiện và hiệu chỉnh về BCD chuẩn

Ta hãy liệt kê những trường hợp IC₁ cho ra tổng S₄S₃S₂S₁S₀ lớn hơn số 01001 (các mã cấm trong BCD) trên bảng 3.8 dưới đây với X là kết quả của mọi trường hợp đã kê: X = 1 khi S₄S₃S₂S₁S₀ > 01001 và X = 0 khi ngược lại. Như vậy X = 1 khi thỏa mãn 1 trong các điều kiện:

- Khi S₄ = 1
- Khi S₃ = 1 và S₂ = 1 hoặc S₂ = S₁ = 1

Như vậy hàm logic của X có dạng:

$$X = S_4 + S_3(S_2 + S_1)$$

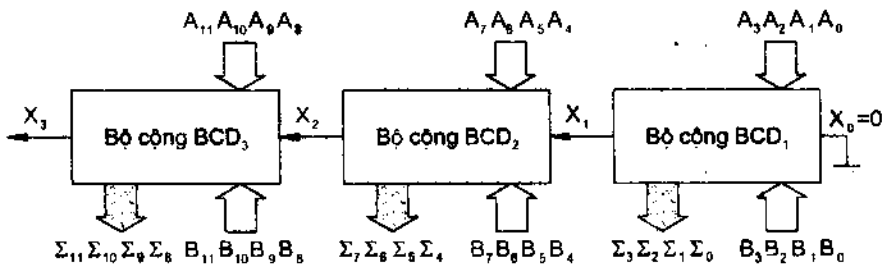
Bảng 3.8

S_4	S_3	S_2	S_1	S_0	X
0	1	0	0	1	0 (9)
0	1	0	1	0	1 (10)
0	1	0	1	1	1 (11)
0	1	1	0	0	1 (12)
0	1	1	0	1	1 (13)
0	1	1	1	0	1 (14)
0	1	1	1	1	1 (15)
1	0	0	0	0	1 (16)
1	0	0	0	1	1 (17)
1	0	0	1	0	1 (18)

Các cổng G_1, G_2 và G_3 có nhiệm vụ tạo hàm X chính là tín hiệu nhớ gửi tới bộ cộng BCD 8421 ở cấp thập phân cao hơn kế tiếp. Tất cả các tập hợp giá trị đầu vào của IC₁ $S_4S_3S_2S_1S_0 \leq 01001$ đều cho $X = 0$ và do vậy bộ cộng dùng IC₂ có các trạng thái của số hạng $A_{32}A_{22}A_{12}A_{02} = 0000$, kết quả là tại đầu ra của IC₂ ta nhận được giá trị của tổng $\Sigma_3 \Sigma_2 \Sigma_1 \Sigma_0 = B_{32}B_{22}B_{12}B_{02} = S_3S_2S_1S_0$ (C_{02} luôn bằng 0).

3. Khi cần cộng hai số BCD 8421 có nhiều cấp thập phân, có thể ghép liên tiếp các mạch hình 3.38 cho mỗi nhóm BCD 4 bit để mở rộng cho số BCD nhiều bit.

Ta xét một ví dụ sau: Cho $A = (A_{11} \dots A_0)$ BCD 8421
 $B = (B_{11} \dots B_0)$ BCD 8421



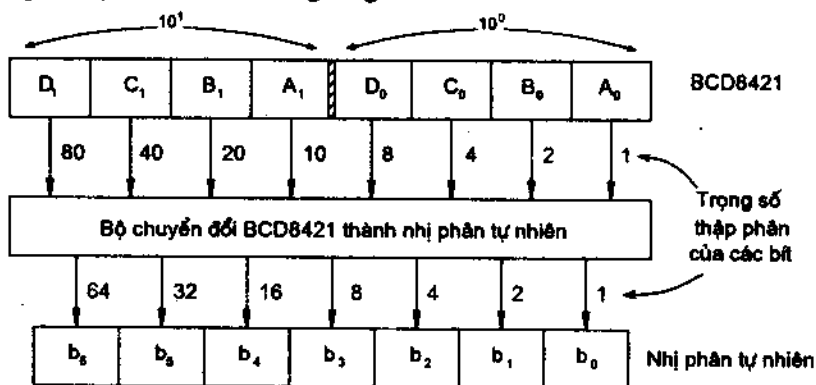
Hình 3.39. Cách nối 3 bộ cộng BCD 4 bit để mở rộng cho phép cộng 2 số A, B ở dạng 12 bit BCD (biểu diễn 3 cấp thập phân tương ứng)

là các số nhị thập phân (BCD) biểu diễn 3 ký số thập phân hàng trăm, hàng chục và hàng đơn vị. Hãy xây dựng bộ cộng thực hiện $A + B$.

Để thực hiện phép cộng cần 3 bộ cộng BCD 4 bit hình 3.38 được ghép nối liên tiếp như hình 3.39 với tổng kết quả là $(X_3 \Sigma_{11} \dots \Sigma_0)_{\text{BCD 8421}}$.

3.7.8. Chuyển đổi số nhị phân BCD 8421 sang số nhị phân tự nhiên tương đương dùng bộ cộng song song 4 bit

1. Sơ đồ hình 3.40 mô tả quá trình chuyển đổi 1 số BCD 8421 dạng $D_1C_1B_1A_1D_0C_0B_0A_0$ thành 1 số nhị phân tương đương $b_6b_5b_4b_3b_2b_1b_0$ với các trọng số vị trí các bit tương ứng.



Hình 3.40. Quá trình chuyển đổi BCD 8421 thành nhị phân tự nhiên

2. Bảng 3.9 cho tương ứng trọng số thập phân của mỗi bit trong số BCD 8421 với biểu diễn nhị phân của các trọng số này.

Bảng 3.9

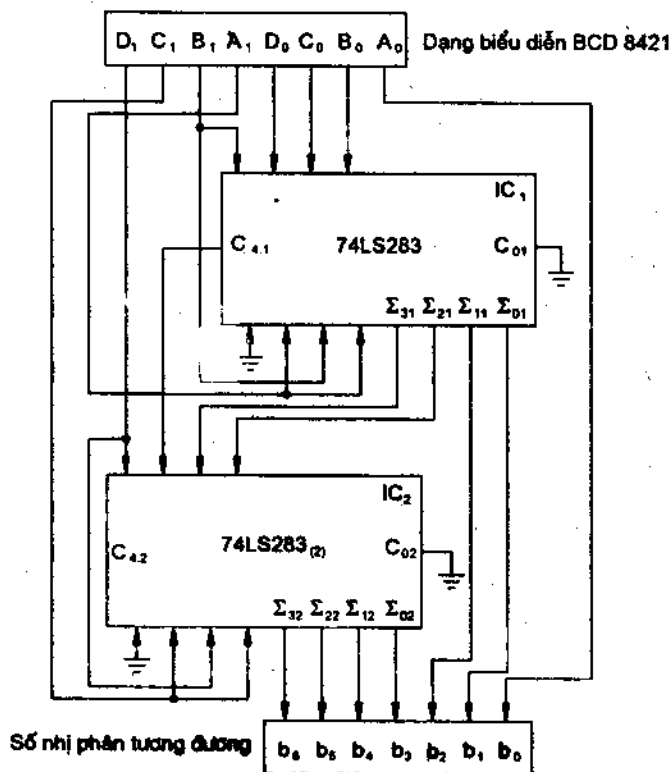
Bit BCD	Trọng số thập phân	Mã nhị phân tương đương						
		b_6	b_5	b_4	b_3	b_2	b_1	b_0
A_0	1	0	0	0	0	0	0	1
B_0	2	0	0	0	0	0	1	0
C_0	4	0	0	0	0	1	0	0
D_0	8	0	0	0	1	0	0	0
A_1	10	0	0	0	1	0	1	0
B_1	20	0	0	1	0	1	0	0
C_1	40	0	1	0	1	0	0	0
D_1	80	1	0	1	0	0	0	0

Từ bảng 3.9 cách chuyển đổi số BCD 8421 thành số nhị phân tự nhiên được thực hiện như sau : "Tính tổng nhị phân của các số nhị phân tương đương với mọi bit có giá trị 1 trong mã BCD".

Ví dụ: Đổi số $(54)_{10} = (01010100)_{\text{BCD 8421}}$ sang số nhị phân , tiến hành theo quy tắc trên có:

$$\begin{array}{r}
 (54)_{10} = 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \\
 \begin{array}{l}
 \text{---} (4)_{10} \rightarrow 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \\
 \text{---} (10)_{10} \rightarrow 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\
 \text{---} (40)_{10} \rightarrow 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0
 \end{array} \\
 \hline
 = (0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0)_2
 \end{array}$$

3. Mạch biến đổi số BCD thành nhị phân dùng bộ cộng song song (hình 3.41)



Hình 3.41. Bộ biến đổi mã BCD 8421 nhị phân dùng bộ cộng song song 4 bit

IC_1 và IC_2 thực hiện phép cộng các tổ hợp bit BCD 8421 theo bảng 3.9:
 $A_0 = b_0$, $B_0 + A_1 = b_1$ (bỏ qua C_{42}), $B_1 + C_0 = b_2$ (bỏ qua C_{41}), $D_0 + A_1 + C_1 = b_3$, $(D_0 + A_1)$ ở IC_1 ra Σ_{21} cộng tiếp với C_1 ở IC_2 ra $b_3 = \Sigma_{02} \dots$ cho tới bit MSB : $b_6 = \Sigma_{32} = D_1$.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 3

3.1. Bộ mã hoá và bộ giải mã khác nhau như thế nào?

3.2. Bộ mã hoá ưu tiên khác với bộ mã hoá thông thường ở điểm nào? Trong hình 3.6 nếu người sử dụng bàn phím nhấn cùng một lúc 3 chuyển mạch SW_7 , SW_5 và SW_2 thì điều gì xảy ra ở đầu ra? Hãy viết từ mã nhị phân BCD tại đầu ra theo thao tác này (nếu có).

3.3. a) Có thể cho nhiều đầu ra của bộ giải mã lên mức tích cực được không? Vì sao?

b) Có thể tác động lần lượt các bit nhị phân BCD vào các đầu vào của bộ giải mã được không? Vì sao?

c) Người ta cho 1 từ mã $CBA = 1\ 0\ 1$ tác động lần lượt vào các đầu vào tương ứng (C tác động đầu tiên, sau đó 1 chu kỳ bit B tác động và sau 2 chu kỳ bit A tác động) của bộ giải mã 1 trong 8. Hãy xác định trạng thái đầu ra của bộ giải mã trong trường hợp này (giả thiết tốc độ làm việc của bộ giải mã nhanh hơn 1 chu kỳ bit).

3.4. Đầu vào cho phép của bộ giải mã có nhiệm vụ gì? Với bộ giải mã 7442/7445 có thể biến đổi thành bộ giải mã 1 trong 8 với 1 đầu vào cho phép được không? (Hãy vẽ mạch điện khối của IC 7442 để giải thích cách làm).

3.5. IC 7445 và IC 7442 khác nhau ở chỗ nào? Tương tự hỏi với IC 7442 và 74LS42, 74HC42?

3.6. Biết IC 74154 là bộ giải mã 4 thành 16, có hai đầu vào cho phép (tích cực ở mức thấp). IC này phải có bao nhiêu cổng vào và bao nhiêu cổng ra? Có bao nhiêu chân? Hãy vẽ sơ đồ chân của IC đã cho (xem hình B3.27).

3.7. Đặc điểm chính của mạch logic tổ hợp là gì?

Phương pháp phân tích một mạch tổ hợp gồm mấy bước?

Khi cần thiết kế một mạch tổ hợp cần phải tiến hành như thế nào?

3.8. Hãy xây dựng một mạch logic thực hiện việc lấy biểu quyết đa số: gồm 3 đầu vào và 1 đầu ra: tín hiệu (trạng thái) của đầu ra luôn có mức theo đa số mức của tín hiệu vào.

a) Xây dựng bảng chân lý.

b) Tìm hàm logic đã tối thiểu.

c) Xây dựng mạch điện chỉ dùng các cổng NAND 2 đầu vào IC 7400 (hay IC 4011).

3.9. Trình bày phương pháp biến đổi mã BCD 8421 thành nhị phân áp dụng cho số : $(100101100101) = (965)_{10}$.

3.10. a) Hãy xây dựng một mạch tổ hợp chuyển từ mã nhị phân 8421 4 bit sang mã vòng Gray ? (xem bảng 1.3 chương 1)

b) Câu hỏi tương tự như câu a với đầu ra là mã thừa 3 (xem bảng chân lý ở 1.3).

3.11*. Hãy xây dựng một mạch tổ hợp có đầu vào là số nhị phân 4 bit DCBA, đầu ra Z. Khi DCBA chia hết cho 3 thì $Z = 1$ các trường hợp khác thì $Z = 0$.

3.12*. Hãy xây dựng một bộ mã hoá đầu vào là các số thập phân từ 0 đến 9, đầu ra là mã vòng thừa 3.

3.13. a) Kết luận rằng "ứng với mỗi tổ hợp mã vào BCD sẽ có một đầu ra của bộ giải mã BCD thành 7 vạch được chuyển đến trạng thái tích cực" có đúng không?

b) Hãy thể hiện các ký số sáng từ 0 đến 9 nhờ các vạch LED tương ứng trong trường hợp các đoạn LED được nối chung katốt đến 0V.

c) Hãy xác định các trạng thái ra của IC 7447 khi có các tổ hợp mã đầu vào tác động là 0 1 0 1, 1 0 0 0, 0 0 1 1 và 1 0 1 0, nhận xét gì với từ mã vào cuối cùng?

3.14. Các đoạn LED anốt chung có dòng làm việc là 15mA, điện áp thuận rơi trên LED lúc sáng là 2,4V. Hãy xác định giá trị điện trở hạn dòng R để đảm bảo điều kiện sáng bình thường của LED.

a) Khi chọn R lớn lên thì điều gì xảy ra? R nhỏ đi?

b) Tính dòng tối thiểu và dòng tối đa nguồn V_{cc} cần nuôi LED khi chúng lần lượt hiển thị sáng các ký số thập phân?

3.15*. Để hiển thị các ký tự A, B, C, D, E, F trong hệ đếm 16 sử dụng LED 7 đoạn cần cho các đầu vào chấp nhận các tổ hợp mã vào từ 1010 đến 1111 (xem bảng 1.1 chương 1).

a) Hãy thiết lập bảng chân lý cho 6 ký tự này ở lối ra màn hình LED anốt chung ứng với các tổ hợp mã vào tương ứng.

b) Viết các hàm ra cho bộ giải mã này.

(Chú ý ta sử dụng 4 biến BCD 8421 đầu vào là $X_3(2^3)$, X_2 , X_1 , $X_0(2^0)$ để tránh lẫn với hàm ra các ký tự A,B,C,D,E,F)

3.16. a) Bộ dồn kênh có nhiệm vụ gì ? Chức năng của đầu vào chọn lựa của nó?

b) Một bộ dồn kênh có thể chuyển được 1 trong số 32 đầu vào dữ liệu đặt tới nó tới đầu ra vậy tổng số bộ MUX sẽ có bao nhiêu đầu vào ? Nếu nhiệm vụ của các đầu vào vừa liệt kê.

c*) Nếu thực hiện bộ dồn kênh 1 trong 32, như vừa nói ở câu (b) bằng các vi điện tử loại 74151 với 8 đầu vào dữ liệu thì phải làm như thế nào ? (dựa theo cách đã làm với hình 3.18 của MUX 16 đầu vào).

3.17*. Có 2 dữ liệu gồm mỗi dữ liệu 8 bit cần qua kiểm soát của một bộ dồn kênh trên đường truyền của chúng, phải làm thế nào để tại những thời điểm nhất định tại cổng ra của MUX sẽ xuất hiện chỉ một trong 2 dữ liệu này. Cấu trúc sẽ gồm bao nhiêu đầu vào dữ liệu? bao nhiêu đầu ra và bao nhiêu đầu vào chọn ?

3.18. a) Hãy giải thích điểm khác biệt giữa MUX và DEMUX

"Mạch của DEMUX giống hệt mạch của bộ giải mã". Đúng hay sai?

b) Hãy xây dựng cấu trúc của DEMUX vào 1 đường, ra 4 đường.

– Trường hợp sử dụng IC 74LS 138.

– Trường hợp dùng các cổng cơ bản.

3.19. a) Bộ giải mã 7442 không có đầu vào cho phép E, sử dụng nó như 1 bộ giải mã 1 trong 8 được không? (nếu không dùng $\bar{Y}_9$ và $\bar{Y}_8$) có thể dùng đầu vào dữ liệu D làm đầu vào cho phép được không?

b) Khi dùng 7442 làm DEMUX 1 ra 8 thì phải sử dụng các đầu vào dữ liệu và đầu vào chọn như thế nào?

3.20. Hãy mô tả quá trình so sánh 2 số nhị phân sau:

a) $A = 10010010$

$B = 10100110$

b) $A = 10100101$

$B = 10101001$

3.21. Khi cần so sánh 2 số nhị phân 16 bit phải dùng mấy IC 7485. Hãy mô tả và thực hiện ghép các IC đã chọn để thực hiện.

Hãy áp dụng để so sánh 2 số nhị phân sau:

$A = 1001\ 0010\ 1101\ 1110$

$B = 1001\ 0010\ 1011\ 0101$

3.22. a) Hãy viết lại bảng chân lý và cấu trúc mạch điện bộ bán tổng.

b) Bộ cộng đầy đủ (FA) khác bộ bán tổng như thế nào? Viết bảng chân lý và giải thích ý nghĩa các biến và hàm trong bảng.

c) Hãy chứng minh hệ thức (3.8) đối với các hàm ra của FA.

3.23. Cho 2 số nhị phân $A = 1010\ 0111$

$B = 1100\ 1010$

a) Hãy xây dựng bộ cộng song song dùng 8 bộ FA từ FA₀ đến FA₇ và 2 thanh ghi 8 bit (xem thêm phần 4.5.3 về thanh ghi)

b) Viết giá trị hàm từ S₇ đến S₀ của đầu ra.

Viết các giá trị nhớ C₇ đến C₀ ở các đầu vào nhớ của các FA từ FA₀ đến FA₇.

c) Hãy viết kết quả cuối cùng của tổng S và thử lại trong phép cộng thập phân. Để biểu diễn S cần mấy bit kết quả?

3.24. a) Ưu nhược điểm của phương pháp cộng song song các bit nhị phân?

b) Cần bao nhiêu chip IC 74HC283 để thực hiện cộng 2 số nhị phân 20 bit? cộng hai số nhị phân 32 bit?

c) Hãy vẽ mạch thực hiện bộ cộng 32 bit.

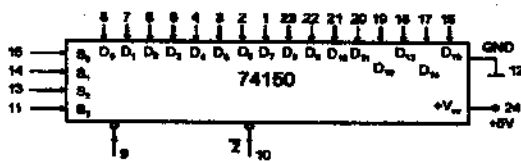
3.25. a) Nếu với mỗi FA gây trễ cho C_1 là $7,5ns$ ($1ns = 10^{-9}s$) thì một chip IC74HC 283 gây trễ cho tín hiệu truyền nhớ là bao nhiêu? Khi thực hiện cộng 32 bit thì thời gian trễ để truyền từ C_0 đến C_{31} là bao nhiêu?

b) Hãy mô tả các trạng thái vào, ra, nhớ của bộ cộng 8 bit hình 3.36b khi số bị cộng là :

$$A = 137_{10} \text{ và số cộng } B = 72_{10}$$

$$C = 245_{10} \text{ và số cộng } D = 168_{10}$$

3.26. Cho vi điện tử MSI có ký hiệu trên hình B3.26.



Hình B3.26

(các số thập phân bên ngoài chỉ vị trí số chân đọc theo quy tắc bố trí cấu tạo xem các hình 2.20 hoặc 2.21 ở chương 2)

a) Hãy mô tả nhiệm vụ và hoạt động của IC đã cho với các ký hiệu:

$\bar{E}$ – tín hiệu điều khiển tích cực thấp. $S_0S_1S_2S_3$ – 4 bit vào chọn địa chỉ.

$D_0 \rightarrow D_{15}$ – các dữ liệu vào.

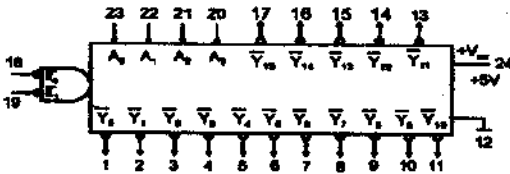
Z – lối ra tích cực thấp.

b) Vẽ mạch logic cấu trúc bên trong của IC.

c) Nếu có thêm bit vào S_4 và 32 dữ

liệu $D_0 \rightarrow D_{31}$, vẽ mạch thực hiện dùng 2 IC đã cho.

3.27 Cho vi điện tử MSI có ký hiệu trên hình B3.27 (số thập phân bên ngoài chỉ vị trí đọc chân).



Hình B3.27

a) Nếu nhiệm vụ và mô tả hoạt động của mạch với :

$\bar{E}_0, \bar{E}_1$ – các tín hiệu điều khiển vô.

$A_0A_1A_2A_3$ – các bit vào địa chỉ.

$\bar{Y}_0 \rightarrow \bar{Y}_{15}$ – các lối ra ở mức tích

cực thấp.

b) Vẽ mạch điện cấu tạo của IC từ các cổng logic cơ bản (AND, OR, NAND, NOR, NOT).

c) Để chọn được 1 trong 64 đầu ra $\bar{Y}_0 \rightarrow \bar{Y}_{63}$ cần dùng mấy IC đã cho?

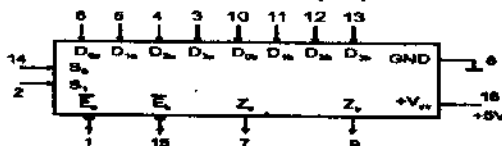
Khi đó cần dùng bao nhiêu bit vào địa chỉ ? Hãy vẽ cách tổ chức thực hiện (chú ý cần bổ sung các cổng logic cơ bản thích hợp để tạo tín hiệu điều khiển chọn vô $\bar{E}_0, \bar{E}_1$ cho từng IC).

3.28. Cho IC 74LS153 ký hiệu trên hình B3.28 với $D_{0n} \dots D_{3n}$ và $D_{0b} \dots D_{3b}$ là hai nhóm dữ liệu vào 4 bit.

Z_a, Z_b là 2 lối ra tương ứng.

S_0, S_1 – các bit địa chỉ.

$\bar{E}_a, \bar{E}_b$ – tín hiệu chọn nhóm.



Hình B3.28

a) Mô tả hoạt động và chức năng của IC đã cho. Viết bảng điều khiển theo mô tả

trên.

b) Vẽ mạch logic cấu tạo bên trong của IC phù hợp với các chức năng đã mô tả.

3.29*. Hãy dùng 9IC 74LS138 để xây dựng 1 bộ giải mã 1 trong 64 với 6 bit nhị phân vào $A_5A_4A_3A_2A_1A_0$ mà không dùng thêm bất kỳ một cổng logic nào khác.

Chương 4

TRIGƠ SỐ VÀ PHẦN TỬ LOGIC DẪY

4.1. KHÁI NIỆM VÀ CẤU TRÚC CƠ BẢN CỦA TRIGƠ SỐ

4.1.1. Mạch Trigơ số (có tên là FlipFlop, viết tắt là FF) là một phần tử logic tuần tự hết sức quan trọng và cơ bản trong kỹ thuật điện tử số. Một FF thường có 2 đầu ra và có ít nhất hai đầu vào thực hiện các chức năng tối thiểu sau:

1. Có hai trạng thái ra là Q và $\bar{Q}$ ổn định và có tính liên hợp, khi $Q = 1$ thì $\bar{Q} = 0$ và ngược lại khi $Q = 0$ thì $\bar{Q} = 1$.

2. Các trạng thái đầu ra không những phụ thuộc vào các trạng thái đầu vào áp đến ngay lúc đang xét mà còn phụ thuộc vào trạng thái quá khứ của mình (trước khi áp tín hiệu vào, Q đang ở trạng thái nào?)

3. Có khả năng tiếp nhận (ghi vào), lưu giữ trong một thời gian tùy yêu cầu và đưa ra (đọc ra) một trạng thái (1bít) nhị phân đang được lưu giữ trong FF.

4. Việc mở rộng chức năng của FF sẽ được thực hiện nhờ có các đầu vào điều khiển hành vi của FF ngoài các đầu vào tín hiệu đã nói tới (xóa tín hiệu đang lưu, cho phép ghi hay đọc...). Có thể sử dụng các cách sau để thực hiện chức năng của một FF:

a) Áp tới các lối vào (gọi là lối vào dữ liệu) mức điện thế thích hợp với mức logic 0 hay mức logic 1 – đây là loại FF làm việc ở chế độ không đồng bộ.

b) Đồng thời với việc trên cần có một lối vào đặc biệt (gọi là đầu vào nhịp) để đưa tới 1 tín hiệu điều khiển (ở mức 1 hoặc mức 0 hoặc lúc ở mức 1 về mức 0 hay lúc ở mức 0 lên mức 1). Chỉ khi xuất hiện tín hiệu điều khiển này (còn được gọi là tín hiệu đồng bộ) khả năng làm việc của FF mới được thực hiện với các đầu vào dữ liệu, đây là loại FF làm việc ở chế độ đồng bộ.

c) Chỉ sử dụng 1 trong số 2 đầu vào đặc biệt gọi là đầu vào điều khiển trực tiếp ghi hay xóa (xem hình 4.3) mà không dùng hai cách đã nêu trên.

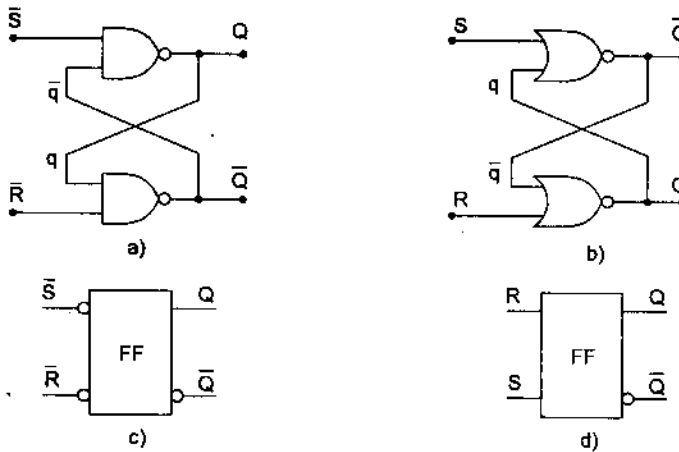
4.1.2. Cấu trúc cơ bản của FF loại RS không đồng bộ

– Tồn tại hai cấu trúc cơ bản của FF (Trigơ RS không đồng bộ) cho trên hình 4.1a và b với các ký hiệu quy ước trên hình 4.1c và d tương ứng.

– Mạch hình 4.1a gồm 2 cổng NAND mắc liên hợp có chứa hai vòng hồi tiếp dương: đầu ra của cổng này nối trực tiếp với một trong hai đầu vào của cổng kia. Tương tự với cấu trúc hình 4.1b cũng chứa hai vòng hồi tiếp dương. Cách nối này minh họa thuộc tính thứ 2 đã nói ở mục (4.1.1) là trạng thái đầu ra tương lai sẽ phụ thuộc vào trạng thái của đầu vào tín hiệu áp tới và phụ thuộc cả vào trạng thái hiện tại của Q (đang ở mức nào).

– Hệ hàm ra của cấu trúc 4.1a và 4.1b tương ứng có dạng biểu thức (4.1):

$$\begin{aligned} Q &= \overline{\overline{S}.q} = S + q & \text{và} & & \overline{Q} &= \overline{\overline{S}+q} = \overline{S}. \overline{q} \\ \overline{Q} &= \overline{\overline{R}.q} = R + \overline{q} & \text{và} & & Q &= \overline{\overline{R}+\overline{q}} = \overline{R}. q \end{aligned} \quad (4.1)$$

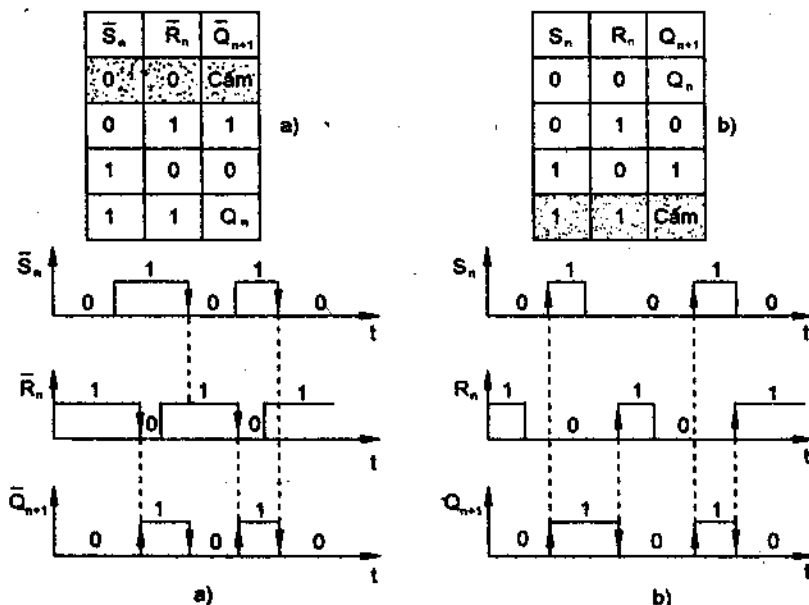


Hình 4.1. Mạch điện RSFF không đồng bộ
a, c) Dùng 2 cổng NAND ; b, d) Dùng 2 cổng NOR.

Bảng trạng thái (chức năng) và đồ thị thời gian minh họa hoạt động của các mạch hình 4.1 cho trên hình 4.2.

Để hệ các hàm ra (4.1) đơn trị đảm bảo thuộc tính thứ nhất đã nêu ở (4.1.1), yêu cầu ở cổng (4.1a) có cấu trúc từ 2 NAND hay cổng (4.1b) có cấu trúc từ 2 NOR không được phép đồng thời áp vào hai tín hiệu cùng ở mức tích cực, tức là phải đảm bảo điều kiện $R_n. S_n = 0$.

Đồ thị hình 4.2c và d dẫn tới các kết luận là:



Hình 4.2. Bảng chức năng và đồ thị xung thời gian minh họa chức năng (chỉ số n chỉ nhịp hiện tại ; $n + 1$ là nhịp tương lai)
a) Bảng chức năng của FF từ 2 NAND ; b) Bảng chức năng của FF từ 2 NOR
Giản đồ điện thế tín hiệu xung minh họa hoạt động của RSFF không đồng bộ có cấu trúc từ 2 cổng NAND (c) và từ hai cổng NOR (d)

1. RSFF có cấu trúc từ 2 cổng NAND chỉ chuyển trạng thái tại lúc được kích hoạt bằng mức tích cực thấp (khi tín hiệu vào chuyển xuống thấp, lưu ý các biến đầu vào là $\bar{S}_n$ và $\bar{R}_n$).

2. RSFF có cấu trúc từ 2 cổng NOR chỉ chuyển trạng thái tại lúc được tác động tới đầu vào tín hiệu ở mức tích cực cao (từ 0 lên 1).

3. Mỗi khi đầu vào R_n được áp lên mức 1, Q_{n+1} xuống mức thấp, còn mỗi khi S_n áp lên 1, Q_{n+1} lên mức cao (xem hình 4.2d).

4. Việc trở về 0 của R_n và S_n không ảnh hưởng tới trạng thái ra. Từ bảng trạng thái, có thể viết lại hàm ra dạng sau:

$$\left. \begin{aligned} Q_{n+1} &= Q_n \cdot \bar{R}_n \cdot \bar{S}_n + \bar{R}_n S_n \\ R_n \cdot S_n &= 0 \end{aligned} \right\} \quad (4.2)$$

Cộng từng vế hai hệ thức 4.2 và biến đổi sẽ nhận được phương trình đặc tính của RSFF (không đồng bộ).

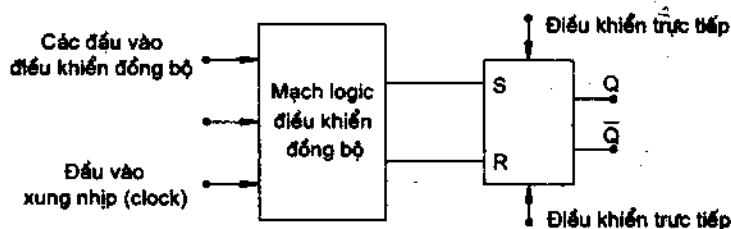
$$Q_{n+1} = S_n + \bar{R}_n Q_n \quad (4.3)$$

Đầu vào S được gọi tên là đầu vào thiết lập (Set), đầu vào R được gọi tên là đầu vào xóa (Reset).

– Hai dạng RSFF (không đồng bộ) đã xét trên có khả năng thực hiện chức năng của một ô nhớ một ký số nhị phân vì có thể duy trì trạng thái ổn định tại các đầu ra lâu tùy ý nếu không có tín hiệu R hoặc S tác động tới đầu vào. Trên thực tế chúng ít có lợi nếu dùng trực tiếp các đầu vào như vậy vì hai lý do:

1. Không biết trước được lúc khởi động (cấp nguồn 1 chiều) FF đang ở trạng thái nào (đang ở mức cao hay mức thấp?).

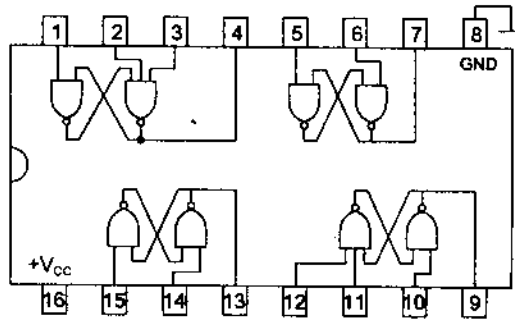
2. Cần phải kiểm soát được trạng thái hiện tại của FF và làm thay đổi được trạng thái (đến tương lai) của nó theo ý muốn trong một mối liên hệ logic với các phần tử số khác của một hệ thống số. Vì vậy các phần tử FF hiện tại đều có bổ sung phần điều khiển gồm 2 loại : điều khiển theo 1 nhịp chuẩn đã thống nhất xác định (gọi là điều khiển đồng bộ) hoặc điều khiển trực tiếp như mô tả trên hình 4.3.



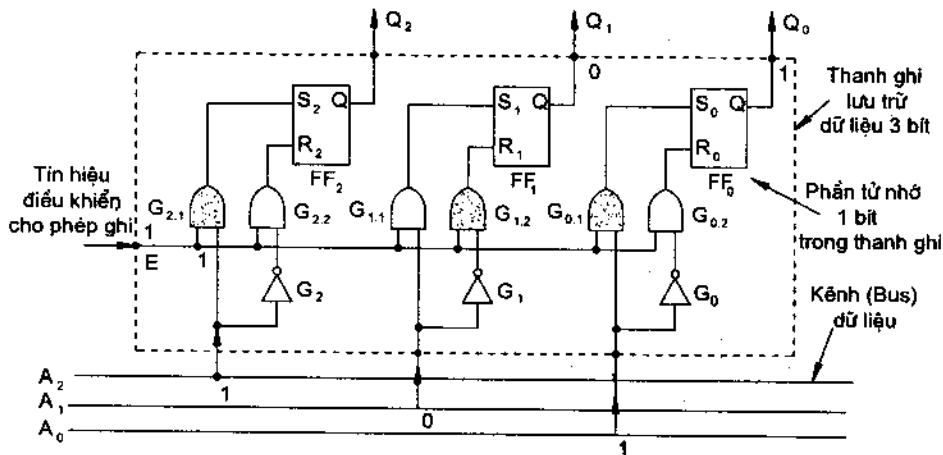
Hình 4.3. Các đầu vào điều khiển đồng bộ và điều khiển trực tiếp RSFF

4.1.3. Ví dụ ứng dụng RSFF không đồng bộ

Hình 4.4 là sơ đồ bố trí chân của IC 74279 có cấu tạo bên trong gồm 4 RSFF không đồng bộ loại NAND. Hình 4.5 minh họa một ứng dụng của chúng khi sử dụng 3 RSFF tổ chức một thanh lưu giữ (thanh ghi- Register) 3 bit nhị phân dữ liệu trong một thời gian nào đó (khi chưa sử dụng các dữ liệu này ngay lúc đó).



Hình 4.4. Sơ đồ bố trí chân của IC 74279 (RSFF không đồng bộ)

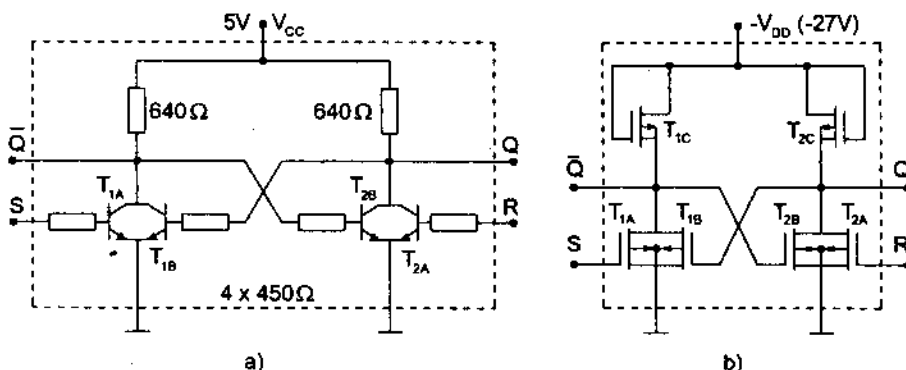


Hình 4.5. Ứng dụng RSFF khi sử dụng 3 RSFF

Dữ liệu $A_2A_1A_0$ trên kênh dữ liệu sẽ được ghi vào các RSFF, nghĩa là khi $E = 1$ nhóm $G_{2.1}$, $G_{1.2}$ và $G_{0.1}$ sẵn sàng cho phép $A_2=1$, $A_1=0$ và $A_0=1$ áp vào S_2, S_1, S_0 tương ứng của FF_2 , FF_1 và FF_0 , trong khi đó nhóm $G_{2.2}$, $G_{1.1}$ và $G_{0.2}$ bị các cổng G_2 , G_1 và dữ liệu $A_1 = 0$ cho tín hiệu khóa lại, cổng S_2 , S_1 , S_0 ở mức tích cực và ta có $Q_2 = A_2 = 1$, $Q_1 = A_1 = 0$, $Q_0 = A_0 = 1$, nhóm dữ liệu A_i đã được ghi vào và lưu giữ trong thanh ghi thể hiện ở các trạng thái Q_i . Như vậy, nhóm dữ liệu A_i là 1 0 1 thì chỉ có $Q_2 = A_2 = 1$, $Q_0 = A_0 = 1$. Còn $A_1 = 0$ nên $G_{1.1}$, khóa $G_{1.2}$ mở cho phép $\bar{A}_1$ vào R_1 xóa $Q_1 = 0$ (như ký số nhị

phần thể hiện trên hình 4.5). Khi $E = 0$ thì mọi dữ liệu A_i trên kênh dữ liệu đều bị bỏ qua không được phép thu nhận vào thanh ghi (chế độ bị cấm vì “đã được sử dụng lưu giữ trước đó rồi và vẫn còn đang cần giữ tiếp tục- đã bị ghi đầy” hay “được để dành cho một việc có mức ưu tiên cao hơn” nào đó...). Trong ví dụ trên các RSFF làm việc với các mức tích cực cao của S và R nên chúng là loại có cấu trúc dùng 2 cổng NOR hình 4.1b.

4.1.4. Mạch điện thực hiện RSFF không đồng bộ có thể sử dụng cổng NOR loại RTL (hình 4.6a) hay dùng NAND loại PMOS (hình 4.6b).



Hình 4.6. a) Mạch điện RSFF không đồng bộ dùng các phần tử NOR RTL
b) Mạch RSFF dùng các phần tử PMOS

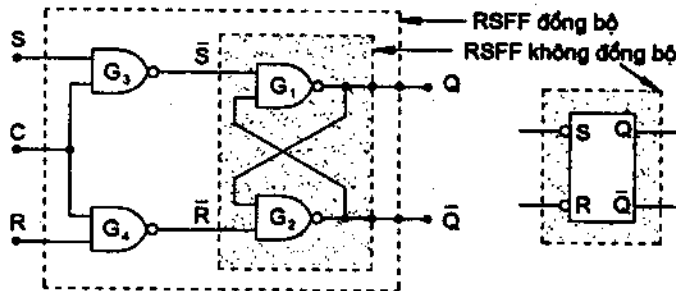
Các cặp tranzito T_1 và T_2 làm việc ở chế độ khóa kiểu bập bênh, nối mạch và ngắt mạch so le nhau. Khi $S = 1$ (hình 4.6a) T_{1A} nối mạch làm mức điện thế colectơ 1 xuống thấp ($\bar{Q} = 0$) dập tắt T_{2B} ; điện thế colectơ của T_2 dâng cao về mức 1 ($Q = 1$), còn khi $R = 1$ tình hình ngược lại T_{2A} nối mạch T_{1B} ngắt và $Q = 0$, $\bar{Q} = 1$. Khi $R = S = 0$, mạch giữ ở trạng thái đang có không thay đổi các mức ra Q và $\bar{Q}$. Khi $R = S = 1$, T_1 và T_2 đều nối mạch dẫn tới $Q = \bar{Q} = 0$ vi phạm điều kiện làm việc so le ở đó các giá trị Q và $\bar{Q}$ liên hợp nhau nên trạng thái này bị cấm đối với các đầu vào điều khiển trực tiếp, thể hiện ở điều kiện cấm ở hệ thức (4.2) : $R.S = 0$.

4.2. RSFF LOẠI ĐỒNG BỘ

4.2.1. RSFF đồng bộ (tĩnh)

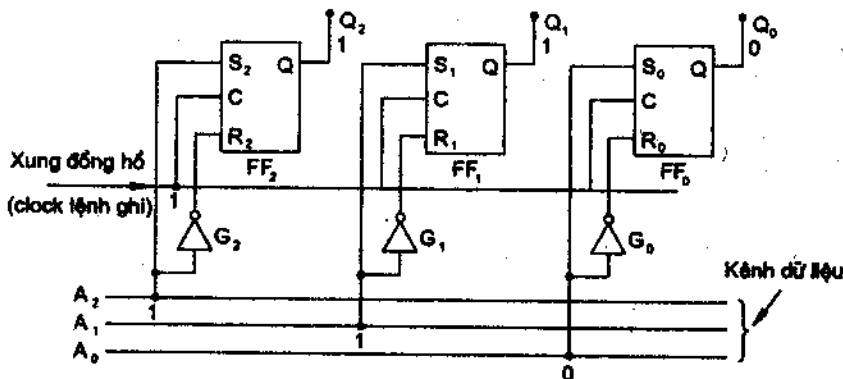
Hình 4.7 giới thiệu mạch RSFF đồng bộ, mạch gồm 4 cổng NAND 2 đầu vào : G_1 G_2 là phần tử RSFF cơ bản ở hình 4.1a ; các cổng G_3 và G_4 được bổ sung nhằm kiểm soát dữ liệu S và R được phép hay không được phép thu nhận

vào RSFF : khi tín hiệu điều khiển (tín hiệu đồng bộ) $C = 1$, G_3 và G_4 nhận lệnh cho phép S hoặc R thâm nhập vào G_1 và G_2 (phải cấm trạng thái $R = S = 1$ theo bảng chân lý quy định), còn khi $C = 0$ các cổng G_3 và G_4 sẽ không cho phép tín hiệu vào R và S áp sát tới G_1 , G_2 tức là RSFF có lệnh không được phép làm việc với các dữ liệu vào; sẽ ở nguyên trạng thái đang có. Tín hiệu điều khiển C thường được gọi là xung đồng hồ (clock - bất nhịp) hay xung chọn mở cổng FF.



Hình 4.7. RSFF đồng bộ tính làm việc theo xung nhịp c

4.2.2. Thanh ghi lưu trữ dữ liệu dùng RSFF đồng bộ (hình 4.8). Mạch điện hình 4.8 tương tự mạch thanh ghi trên hình 4.5 đã xét, ở đây các RSFF không đồng bộ đã được thay thế bằng loại RSFF đồng bộ và nhờ đó các cổng logic điều khiển G_{01} , G_{02} , G_{11} , G_{12} và G_{21} , G_{22} không còn cần thiết và cấu trúc trở nên gọn gàng hơn, mạch có khả năng ghi giữ 3 bit nhị phân.

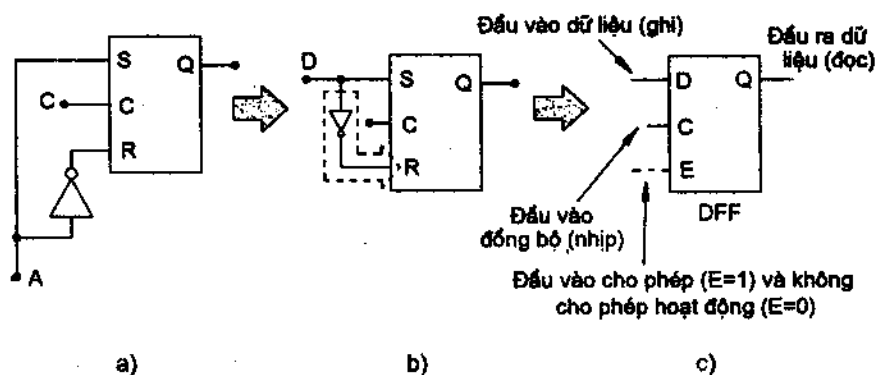


Hình 4.8. Thanh ghi 3 bit dùng RSFF đồng bộ dạng mạch hình 4.7 (khi có xung đồng hồ $C = 1$ mạch đang ghi $A_2, A_1, A_0 = 110$)

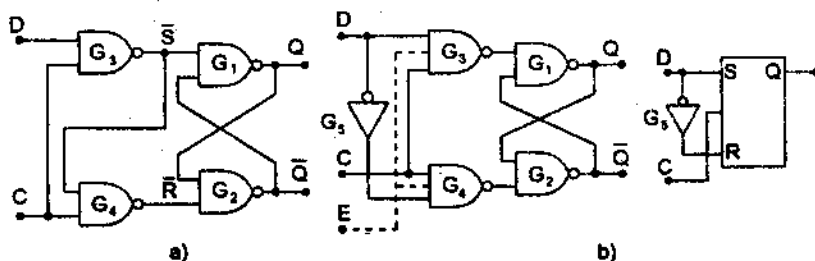
4.3. MẠCH CHỐT DỮ LIỆU (LATCH) - TRIGƠ TRỄ HAY DFF

4.3.1. Bộ chốt dữ liệu 1 bit

Trong mạch điện thanh ghi hình 4.8 vừa xét, nếu đưa các bộ đảo (NOT) $G_2G_1G_0$ vào trong $RSFF_2$, $RSFF_1$ và $RSFF_0$ tương ứng, sẽ nhận được một thanh chốt dữ liệu 3 bit bao gồm ba FF đồng bộ mới còn gọi là các Trơ trễ DFF. Vậy thanh chốt dữ liệu 1 bit có cấu tạo là một RSFF đồng bộ kết hợp với 1 cổng NOT tại trước đầu vào R ; khi cổng NOT được bố trí bên trong RSFF đồng bộ ta nhận được một Trơ trễ DFF hay một thanh chốt dữ liệu 1 bit. Mạch chốt dữ liệu có nhiệm vụ bắt giữ và lưu tín hiệu xuất hiện tại đầu vào của nó, thời gian lưu giữ vào lúc nào, lâu hay mau là do đầu vào nhịp (clock) của nó quyết định.



Hình 4.9. a) Thanh ghi 1 bit (của hình 4.8) ; b) Đặt lại tên đầu vào dữ liệu là D ; c) Sau khi đưa cổng NOT vào trong RSFF đồng bộ (ở phía cửa R) chỉ còn lại cửa D



Hình 4.10. a) Mạch cấu trúc DFF dùng 4 cổng NAND ; b) DFF cấu tạo từ RSFF đồng bộ có bổ sung 1 cổng đảo G_3

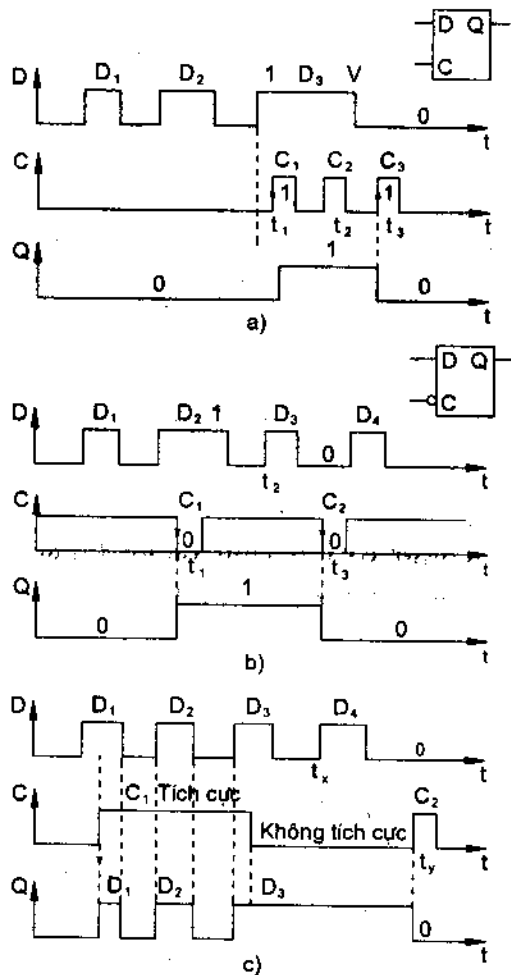
Hình 4.9 mô tả việc hình thành 1 bộ chốt dữ liệu 1 bit (phần tử nhớ có 1 đầu vào dữ liệu).

Khi không có đầu vào nhịp (clock), mạch chốt không đồng bộ, theo hình 4.9b có $S = D$ và $R = \bar{D}$ lúc đó đầu ra Q chỉ tuân theo các thay đổi ở D tức là một bộ lặp lại dữ liệu D không có ý nghĩa trong kỹ thuật số nên trường hợp này ít được sử dụng, đầu vào E có mặt trong một số DFF với chức năng cho phép DFF hoạt động khi $E = 1$.

4.3.2. Mạch điện của DFF dùng các cổng G_3 và G_4

Hình 4.10 là mạch điện của DFF dùng các cổng $G_1G_2G_3$ và G_4 . Mạch là sự bổ sung trước cấu trúc cơ bản hình 4.10a hai cổng G_3 và G_4 hay bổ sung cổng đảo G_3 trước RSFF đồng bộ 4.10b.

- Trường hợp xung clock là mức tích cực cao (hình 4.11a): D_1 và D_2 là dữ liệu đã áp tới đầu vào D nhưng vì $C = 0$ nên DFF bỏ qua, Q giữ nguyên ở mức 0 ban đầu. Khi $t = t_1$ xuất hiện C_1 lên mức tích cực cao, D_3 được DFF tiếp nhận ($Q = 1$) và được lưu giữ lại trong nó, lúc t_2 lại xuất hiện C_2 ở mức tích cực cao nhưng vì Q đã ở 1 nên Q không đổi (lúc này có lưu giữ 1 khác thì cũng thế thôi). Khi C_3 xuất hiện lên mức tích cực cao đồng thời tại t_3 , $D = 0$, khi đó $Q = 0$ (lưu giữ mức 0 của D).



Hình 4.11. Các trường hợp lưu giữ tín hiệu D khi tín hiệu cho phép giữ (nhịp - Clock) có các thuộc tính khác nhau

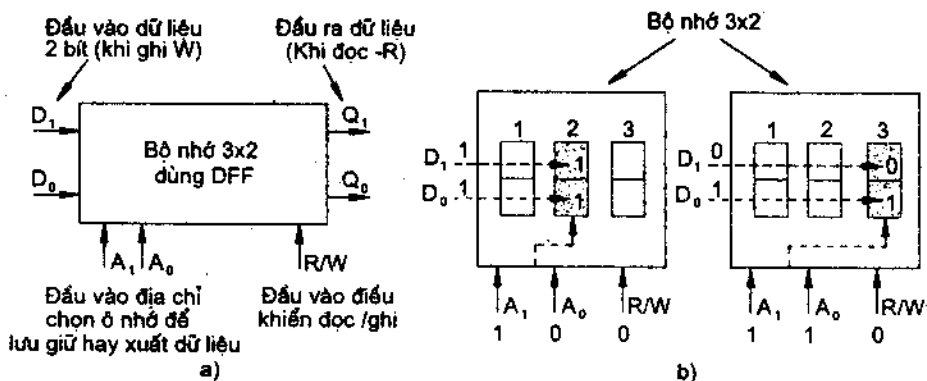
- a) Trường hợp C tích cực ở mức cao
- b) Trường hợp C tích cực ở mức thấp
- c) Trường hợp C tích cực cao có độ rộng khác nhau

– Trường hợp hình 4.11c khi $C = 1$ trạng thái ra lặp lại dạng của D ($Q = D$) lúc $C = 0$ trạng thái Q tại lúc t_x có $D_4 = 1$ nhưng $C = 0$ nên dữ liệu này bị bỏ qua, $Q = 1$ lúc này là do D_3 được lưu giữ từ lúc C hết tích cực chuyển về mức 0, lúc t_y có $C_2 = 1$ (tích cực) $D = 0$ được DFF lưu giữ lại.

2. Mạch chốt dữ liệu dùng làm phần tử nhớ

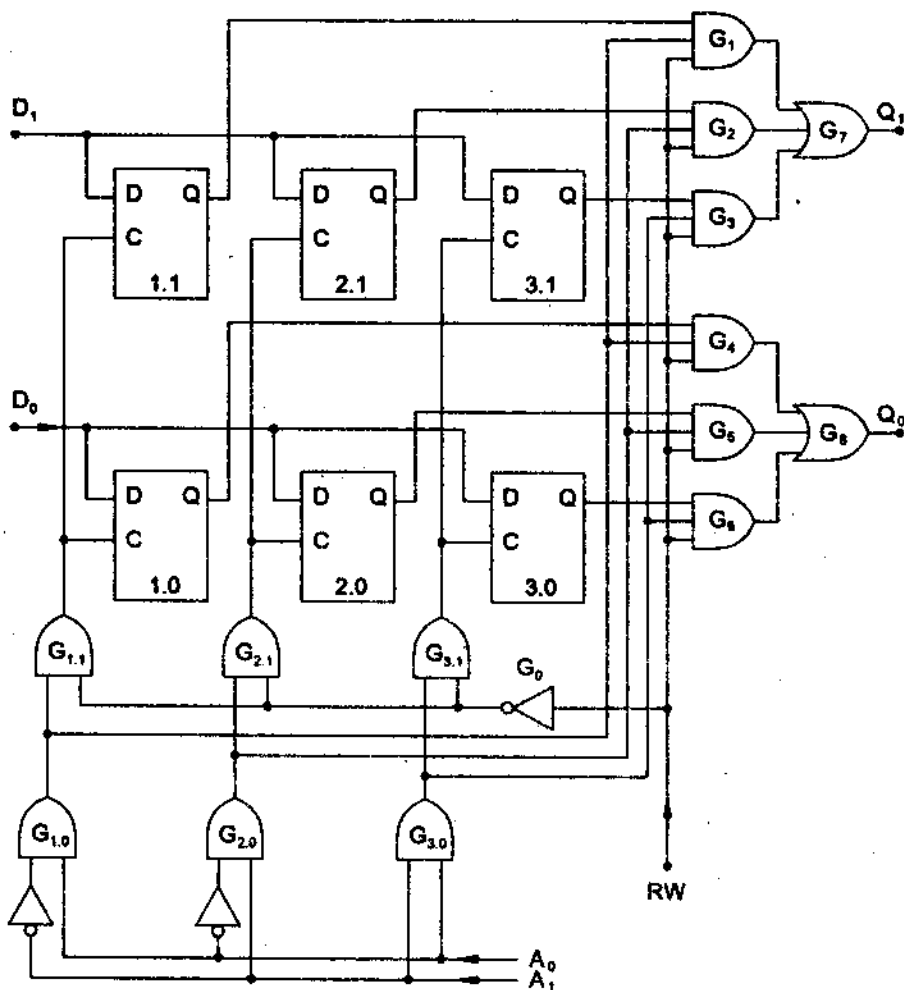
Mạch điện hình 4.13 là một bộ nhớ (3×2) dùng 6DFF có khả năng lưu giữ 3 từ nhị phân 2 bit D_1D_0 vào một vị trí nhớ bất kỳ trong 3 cặp vị trí ô nhớ của nó. Hình 4.13a,b là cấu trúc khối và hình 4.13c là mạch điện chi tiết gồm các đầu vào và đầu ra dữ liệu D_1D_0 , đầu vào địa chỉ A_1A_0 và đầu vào điều khiển R/ W (đọc/ghi).

Với cặp bit A_1A_0 qua $G_{10} G_{20} G_{30}$ tạo ra 3 mã địa chỉ 01, 10 và 11 đặt tới các cổng $G_{11} G_{21}$ và G_{31} lúc đó nếu tín hiệu (R/W) cho phép ở mức tích cực thấp sẽ mở và cho phép xung chọn địa chỉ thích hợp tới các đầu vào C của nhóm DFF cùng chỉ số: ví dụ khi xuất hiện $A_1A_0 = 01$ thì nhóm DFF₁₁ và DFF₁₀ được chọn nhờ đó dữ liệu D_1 và D_0 được ghi và lưu giữ trong chúng, còn khi xuất hiện mã địa chỉ $A_1A_0 = 11$, DFF₃₁ và DFF₃₀ được chọn để ghi D_1D_0 . Trong lúc thực hiện thao tác ghi do (R/W) = 0 nên các cổng G_1 đến G_6 bị cấm, các đầu ra của mọi DFF bị cách ly, dữ liệu D_1D_0 không xuất hiện được ở cổng ra của G_7, G_8 . Khi có lệnh đọc, các cổng G_{11}, G_{21} và G_{31} bị cấm vì tín hiệu (R/W) = 1 (qua bộ đảo G_9) mọi DFF không có tín hiệu vào đầu vào C nên không được phép ghi, các mã địa chỉ A_1A_0 sau khi qua $G_{10} G_{20} G_{30}$ đặt trực tiếp vào nhóm cổng G_1 đến G_6 cho phép thực hiện đọc dữ liệu D_1 và D_0 đang được nhớ tại một trong các ô nhớ thuộc hàng trên và thuộc hàng dưới trong 2 nhóm (DFF₁₁, DFF₂₁ và DFF₃₁ cho dữ liệu D_1 và DFF₁₀, DFF₂₀ và DFF₃₀ cho dữ liệu D_0) nhờ 1 trong 3 cổng $G_1G_2G_3$ được mở cho phép D_1 qua và 1 trong 3 cổng G_4, G_5, G_6 mở cho phép D_0 qua, đặt tới các đầu ra của bộ nhớ $Q_1 = D_1$:



Hình 4.13. a) Sơ đồ khối bộ nhớ 3×2 dùng 6DFF ; b) Đầu vào địa chỉ A_1A_0 hướng dữ liệu D_1D_0 tới vị trí ô nhớ sẽ lưu giữ chúng (khi có lệnh điều khiển W cho phép ghi)

$Q_0 = D_0$, lưu ý rằng tín hiệu $R/W = 1$ đặt tới tất cả các đầu vào của G_1 đến G_6 thể hiện lệnh cho phép đọc đã có. Ví dụ khi $R/W = 1$ với mã địa chỉ $A_1A_0 = 10$: G_2 mở cho phép D_1 đang nhớ ở DFF₂₁ tới đầu ra Q_1 ta nhận được $Q_1 = D_1$ và G_5 mở cho phép D_0 đang nhớ ở DFF₂₀ truyền tới đầu ra $Q_0 = D_0$.



Hình 4.13. c) Sử dụng 6 bộ chốt DFF trong bộ nhớ 3 x 2.

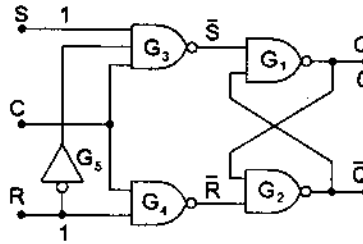
4.4. CÁC DẠNG ĐẶC BIỆT CỦA RSFF

Từ bảng trạng thái hình 4.2a và b, RSFF có một tổ hợp vào $R = S = 1$ bị cấm sử dụng. Người ta chế tạo các loại RSFF biến thể để tận dụng được trạng thái vào cấm này.

4.4.1. Triger R (RSFFR) có bảng chân lý cho trên hình 4.14a và mạch điện cho trên hình 4.14 b.

Vào		Ra
S_n	R_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	0

a)



b)

Hình 4.14

a) Bảng chân lý của RSFF - R ; b) Mạch điện thực hiện RSFF - R

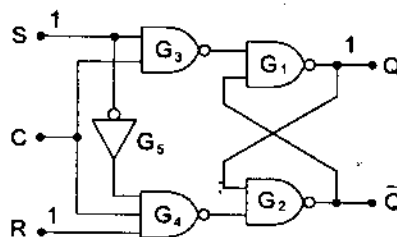
Từ bảng chân lý 4.14a : RSFF-R chấp nhận cả khi $S_n = R_n = 1$ nhờ cổng đảo G_5 nối giữa R với một đầu vào bổ sung của G_3 được tiếp nhận. Tại đầu ra Q ứng với trạng thái $S_n = R_n = 1$ sẽ có giá trị ở mức thấp $Q_{n+1} = 0$.

4.4.2. Triger S (RSFFS) có bảng chân lý và mạch điện cho trên hình 4.15.

Với loại DSFF-S khi tại đầu vào xuất hiện $R_n = S_n = 1$, nhờ cổng đảo G_5 nối từ S tới 1 đầu vào của G_4 (có R áp tới đầu vào khác của G_4) trạng thái ra xác lập ở mức cao $Q_{n+1} = 1$.

Vào		Ra
S_n	R_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	1

a)



b)

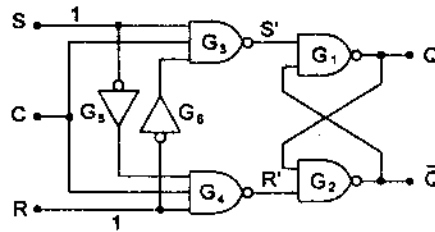
Hình 4.15. RSFF - S

a) Bảng chân lý của RSFF - S ; b) Mạch điện của RSFF - S

4.4.3. Trigo E (RSFF - E) có bảng chân lý và mạch điện thực hiện như trên hình 4.16.

Vào		Ra
S_n	R_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Q_n

a)



b)

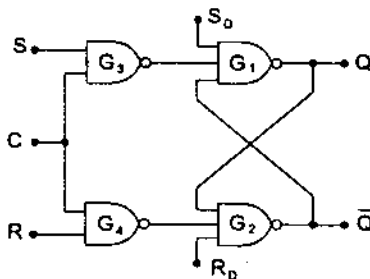
Hình 4.16

a) Bảng chân lý của RSFF - E ;

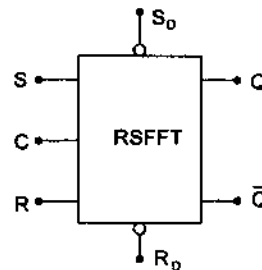
b) Mạch điện thực hiện RSFF - E

Khi các đầu vào xuất hiện tổ hợp $R_n = S_n = 1$, RSFF - E giữ nguyên trạng thái cũ của mình, tức là tương đương với trạng thái $S_n = R_n = 0$ với mức ra duy trì ở trạng thái trước đó nó đang có $Q_{n+1} = Q_n$. Nhờ hai cổng G_5 và G_6 (cổng NOT) các cổng G_3 và G_4 khi $R_n = S_n = 1$ sẽ bị cấm bởi các tín hiệu $\overline{R}_n$ và $\overline{S}_n$, khi đó $S' = R' = 0$ đảm bảo điều kiện $Q_{n+1} = Q_n$.

4.4.4. Trigo T (RSFF - T) còn gọi là RSFF nhịp có mạch điện và ký hiệu quy ước cho trên hình 4.17.



a)



b)

Hình 4.17

a) Mạch điện thực hiện RSFF - T ;

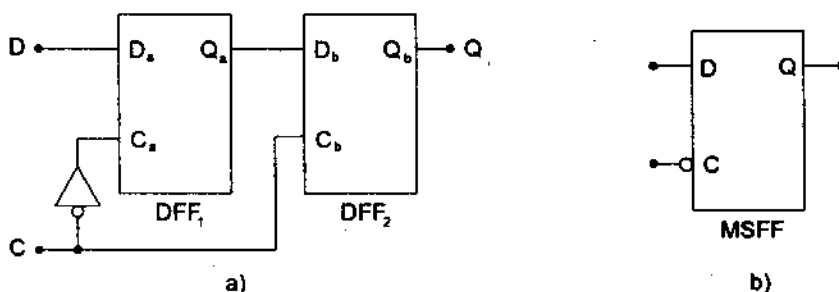
b) Ký hiệu quy ước

Ngoài cấu trúc thông thường với 2 đầu vào đồng bộ S, R được điều khiển theo C như RSFF đồng bộ bình thường, các cổng G_1 , G_2 còn có thêm các đầu vào S_D và R_D điều khiển trực tiếp các trạng thái ra Q khi $C = 0$. S_D và R_D hoạt động giống như các đầu vào S và R với bảng chân lý đã có (cấm $R = S = 1$). Khi $R = S = 0$, có thể thay đổi Q nhờ S_D và R_D .

4.5. TRIGƠ CHÍNH PHỤ MSFF (TRIGƠ CHỦ – TỐ – MASTER SLAVE FLIP FLOP)

Xuất phát từ yêu cầu phần tử nhớ FF có chức năng cùng một lúc khi đang đọc (xuất) giá trị cũ ở đầu ra có thể cho phép nhập (ghi) giá trị dữ liệu mới vào, Trơ chính phụ (MSFF) được xây dựng để đáp ứng đòi hỏi này của một số ô nhớ nhị phân. RSFF thông thường đã khảo sát ở trên không thực hiện được là do chỉ chứa 1 phần tử nhớ ghép chéo (hình 4.1a và 4.1b) giá trị dữ liệu mới được lưu sẽ xuất hiện ngay tại đầu ra của mạch, giá trị lưu trước đó sẽ bị mất ngay khi tiến hành lưu mới bắt đầu được thực hiện.

4.5.1. Trơ chủ tớ có thể xây dựng từ 2 mạch chốt thông thường có cấu trúc cho trên hình 4.18a và ký hiệu quy ước ở dạng 4.18b. Mạch hình 4.18a hoạt động như sau :



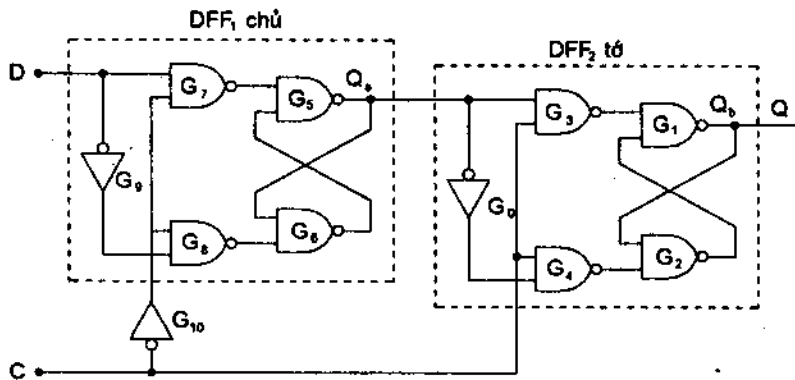
Hình 4.18

a) Kết hợp 2 mạch chốt để tạo MSFF ; b) Ký hiệu quy ước

Khi $C = 1$, DFF₁ bị cấm (vì qua bộ đảo xung nhịp ở mức 0), D không nhập DFF₁, đồng thời DFF₂ được cho phép nhập dữ liệu là mức ra của DFF₁ mức lưu giữ cũ $(Q_a)_n$ vào D_b, tại đầu ra Q xuất hiện $(Q_a)_n$.

Khi $C = 0$, $C_a = 1$, DFF₁ được phép nhập dữ liệu mới D vào nên $(Q_a)_{n+1} = D$, lúc này do $C_b = C = 0$, DFF₂ bị cấm, hai mạch DFF₁ và DFF₂ tách rời nhau. Như vậy nhờ tín hiệu nhịp điều khiển, DFF₁ và DFF₂ làm việc luân phiên nhau : Khi DFF₁ nghỉ thì DFF₂ làm việc (lúc $C = 1$) cho phép dữ liệu đã ghi cũ từ DFF₁ chuyển sang lưu tại DFF₂ và xuất hiện tại đầu ra. Khi DFF₁ làm việc thì DFF₂ nghỉ (lúc $C = 0$) và cho phép dữ liệu mới ghi vào DFF₁. Giá trị mức xung nhịp $C = 1$ mang ý nghĩa là lệnh đọc dữ liệu đã ghi trước. Giá trị

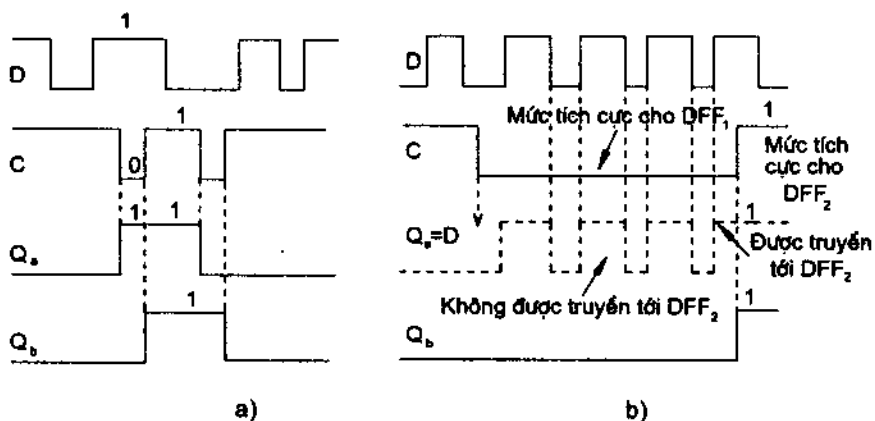
$C = 0$ mang ý nghĩa là lệnh ghi dữ liệu mới. Vị trí cổng đảo quyết định đầu vào C của MSFF tích cực ở mức thấp như ký hiệu quy ước ở hình 4.18b, còn khi nối C trực tiếp vào C_a , qua bộ đảo nối tới C_b ta có MSFF với C tích cực ở mức cao (hình 4.18a).



Hình 4.19. Ghép liên tiếp 2 mạch chốt DFF₁ và DFF₂ để tạo Trơ chủ tớ (MSFF)

Mạch điện logic của MSFF dùng các cổng NAND ghép liên tiếp hai mạch chốt DFF có dạng cho trên hình 4.19 chính là hai mạch hình 4.10b ghép liên tiếp nhau theo cách của hình 4.18a với việc bổ sung thêm cổng đảo G_{10} .

Mạch 4.19 còn được gọi là mạch chốt dữ liệu chủ tớ có đầu vào nhịp clock tích cực ở mức thấp, hoạt động kích bằng mức. Các đồ thị hình 4.20 minh họa hoạt động của MSFF dạng hình 4.19 vừa nêu.

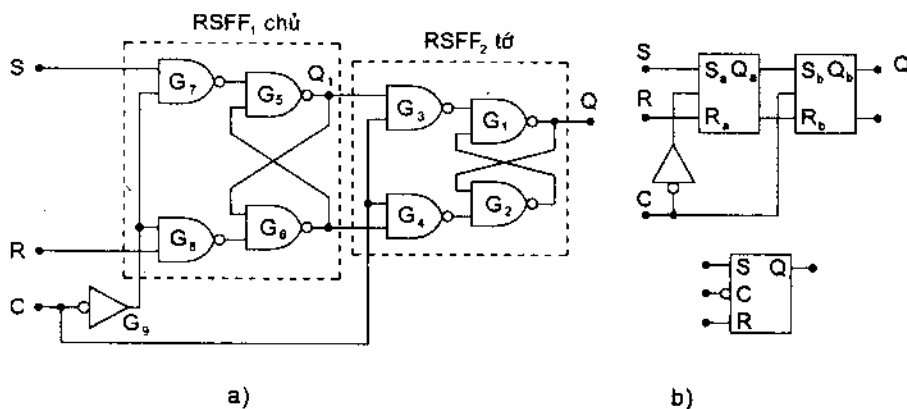


Hình 4.20. Đồ thị thời gian minh họa hoạt động của MSFF dùng DFF₁ và DFF₂
a) Xung nhịp C ngắn so với xung dữ liệu D ; b) Xung nhịp C kéo dài.

4.5.2. MSFF xây dựng từ RSFF

Có thể xuất phát từ 2 RSFF mắc liên tiếp nhau cho làm việc ngược pha nhau nhờ 1 cổng đảo để tạo thành MSFF như thể hiện trên mạch điện hình 4.21a với ký hiệu quy ước cho trên hình 4.21b.

Nguyên tắc hoạt động của mạch hình 4.21a tương tự như mạch 4.19. Khi $C = 0$ mức tích cực kích hoạt RSFF chủ hoạt động cho phép dữ liệu S, R được ghi vào (theo bảng chân lý của RSFF) lúc này RSFF tớ bị cấm và cách ly khỏi RSFF chủ. Khi $C = 1$, RSFF chủ bị cấm, dữ liệu đã ghi tại đầu ra Q_1 của nó



Hình 4.21

a) Mạch điện MSFF từ 2RSFF đồng bộ ; b) Ký hiệu quy ước MSFF nguồn gốc từ RSFF

được phép vào RSFF tớ và xuất hiện ở đầu ra Q. Hai quá trình ghi đọc được phân biệt, dữ liệu đã có trước nhớ trong RSFF chủ được truyền sang RSFF tớ thực hiện thao tác đọc dữ liệu, khi thực hiện thao tác ghi ($C = 0$) dữ liệu mới sẽ được nhớ trong RSFF chủ mà không ảnh hưởng gì tới dữ liệu đã ghi trước đang nhớ ở RSFF tớ.

4.5.3. Một số ứng dụng của MSFF

Các phần tử nhớ dùng MSFF có thể thay thế bình thường ở vị trí các Flip Flop hay mạch chốt dữ liệu thông thường, cấu tạo thành ghi dịch, thực hiện bộ đếm vòng... Trong tiết này sẽ đề cập đến một vài ứng dụng phổ biến của MSFF.

1. Thanh ghi dữ liệu 3 bit dùng MSFF có dạng hình 4.22 tương tự như các thanh ghi lưu trữ dùng DFF hay RSFF đồng bộ đã xét. Khi $\bar{E} = 0$ (cho phép ghi) ; khi $\bar{E} = 1$ (cấm ghi dữ liệu đã ghi từ trước được xuất hiện tại các đầu ra $Q_2Q_1Q_0 = D_2D_1D_0$ hay trạng thái đọc).

2. Thanh ghi dịch chủ yếu là cấu tạo dãy liên tiếp các MSFF chủ yếu dạng hình 4.18 có khả năng vừa ghi dữ liệu mới vừa dịch dữ liệu cũ đã nhớ trong nó đi theo từng nhịp, nếu quá trình dịch xảy ra từ trái sang phải ta gọi là loại dịch phải, thao tác dịch được thực hiện nhờ tín hiệu điều khiển dịch (xung dịch - Shift). Cấu tạo một thanh ghi dịch 4 bit cho trên hình 4.23a, đồ thị thời gian minh họa hoạt động cho trên hình 4.23b để chuyển đổi 4 bit dữ liệu $D_3D_2D_1D_0$ từ dạng nối tiếp sang dạng song song.

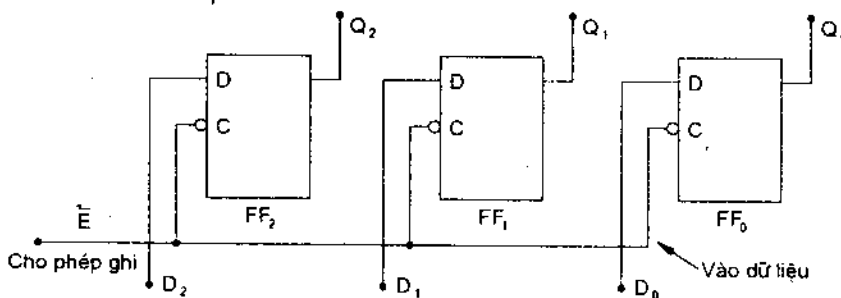
Quá trình biến đổi dãy dữ liệu nối tiếp $D_3D_2D_1D_0 = 1\ 0\ 1\ 1$ thành dữ liệu song song $Q_3Q_2Q_1Q_0 = 1\ 0\ 1\ 1$ dùng thanh ghi dịch hình 4.23a có thể mô tả như sau qua đồ thị thời gian hình 4.23b:

- Vào lúc $t_{(0)}$, $D = 0$, xung dịch $S = 1$ (không tích cực), tất cả các đầu vào và đầu ra đều bằng 0, thanh ghi bị xóa (CLEAR).

- Lúc $t_{(1)}$, có $D_3 = 1$ áp tới FF_0 , đồng thời $S = 0$, $D_3 = 1$ được ghi vào FF_0 đồng thời 0 nhập vào FF_1 , FF_2 và FF_3 từ các đầu ra của FF_0 , FF_1 và FF_0 .

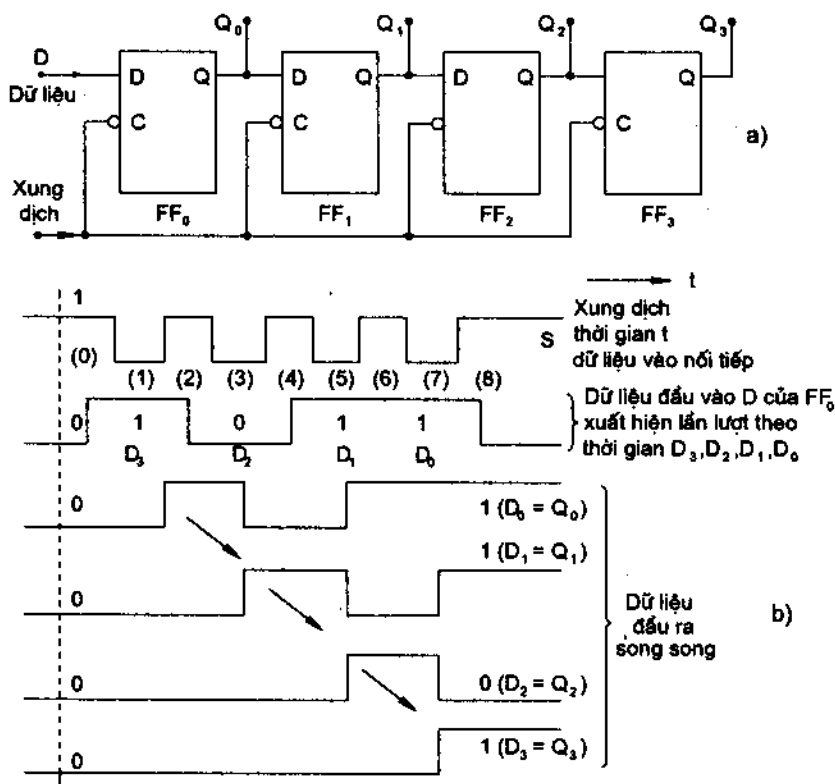
- Lúc $t_{(2)}$, $S = 1$ các giá trị nhập vào FF_i xuất hiện tại đầu ra $Q_3Q_2Q_1Q_0 = 0\ 0\ 0\ 1$.

- Lúc $t_{(3)}$, $S = 0$ (chuyển lên tích cực), $D_2 = 0$ được FF_0 tiếp nhận $D_3 = 1 = Q_0$ được FF_1 tiếp nhận, 0 nhập vào FF_2 và FF_3 từ FF_1 và FF_2 .



Hình 4.22. Thanh ghi lưu trữ 3 bit dữ liệu dùng 3 MSFF

- Lúc $t_{(4)}$, $S = 1$ (không tích cực) các giá trị đã nhập xuất hiện tại đầu ra: $Q_3Q_2Q_1Q_0 = 0\ 0\ 1\ 0$
- Lúc $t_{(5)}$, $S = 0$, $D_1 = 1$ được FF_0 tiếp nhận, $D_2 = 0$ được FF_1 tiếp nhận, $D_3 = 1$ được FF_2 tiếp nhận, 0 từ FF_2 nhập vào FF_3 .
- Lúc $t_{(6)}$, $S = 1$ tại các lối ra có $Q_3Q_2Q_1Q_0 = 0\ 1\ 0\ 1$
- Lúc $t_{(7)}$, $S = 0$, $D_0 = 1$ được FF_0 tiếp nhận, $D_1 = 1$ được FF_1 tiếp nhận, $D_2 = 0$ được FF_2 tiếp nhận và $D_3 = 1$ được FF_3 tiếp nhận.
- Lúc $t_{(8)}$, $S = 1$ các giá trị đã nhập vào, FF_i xuất hiện tại các đầu ra tương ứng, lúc đó $Q_3Q_2Q_1Q_0 = 1\ 0\ 1\ 1 = D_3D_2D_1D_0$. Như vậy sau 4 nhịp tích cực của xung dịch S , toàn bộ 4 bit dữ liệu dạng nối tiếp $D_3D_2D_1D_0$ đã được

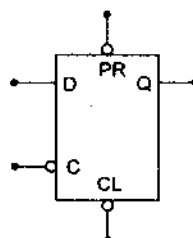


Hình 4.23. a) Cấu tạo thành ghi dịch 4 bit dùng Trơ chủ tử có nguồn gốc từ DFF;
b) Minh họa quá trình ghi dịch phải biến đổi dữ liệu $D_3D_2D_1D_0$ xuất hiện nối tiếp thành xuất hiện đồng thời (song song) sau 4 nhịp ghi - dịch.

ghi vào thanh ghi dịch và chuyển thành 4 bit dữ liệu song song có mặt đồng thời tại các đầu ra của FF_i là $Q_3Q_2Q_1Q_0 = D_3D_2D_1D_0$. Trong trình tự thao tác này, yêu cầu quan trọng là thời điểm nhập dữ liệu và xung dịch ở mức tích cực phải đồng bộ với nhau. Nếu chỉ dùng 1 đầu ra Q_3 sau 8 nhịp tích cực của xung dịch, ta nhận lại dãy nối tiếp ban đầu (biến đổi nối tiếp - nối tiếp).

3. Thanh ghi dịch đa năng là loại có khả năng chuyển đổi dữ liệu ở bốn dạng khác nhau là:

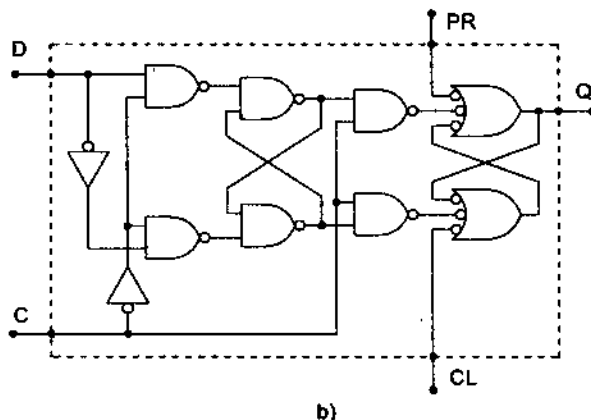
- Vào nối tiếp ra song song
- Vào song song ra nối tiếp
- Vào nối tiếp ra nối tiếp
- Vào song song ra song song.



a)
Hình 4.24a
Mạch chốt đa năng

Các mạch chốt dữ liệu khi đó được bổ sung thêm hai đầu vào điều khiển trực tiếp ở mức tích cực thấp là đầu vào PRESET (PR) và đầu vào CLEAR (CL): khi áp mức 0 vào PR trị 1 được lưu ($Q = 1$), khi áp mức 0 vào CL trị 0 được lưu ($Q = 0$). Hình 4.24a mô tả ký hiệu quy ước của mạch chốt dữ liệu chủ tố có thêm 2 đầu vào mới này và mạch điện của nó được cho trên hình 4.24b có nguồn gốc từ DFF chủ tố đã xét.

Thanh ghi dịch đa năng 4 bit có chức năng nhập dữ liệu song song dùng các đầu vào bổ sung PR và CL được cho trên mạch hình 4.25. Để nhập các giá trị song song, xung vào điều khiển E phải bằng 1. Ví dụ với dữ liệu $B_0B_1B_2B_3 = 0110$ cần nhập đồng thời (song song) vào thanh ghi mạch làm việc như sau:



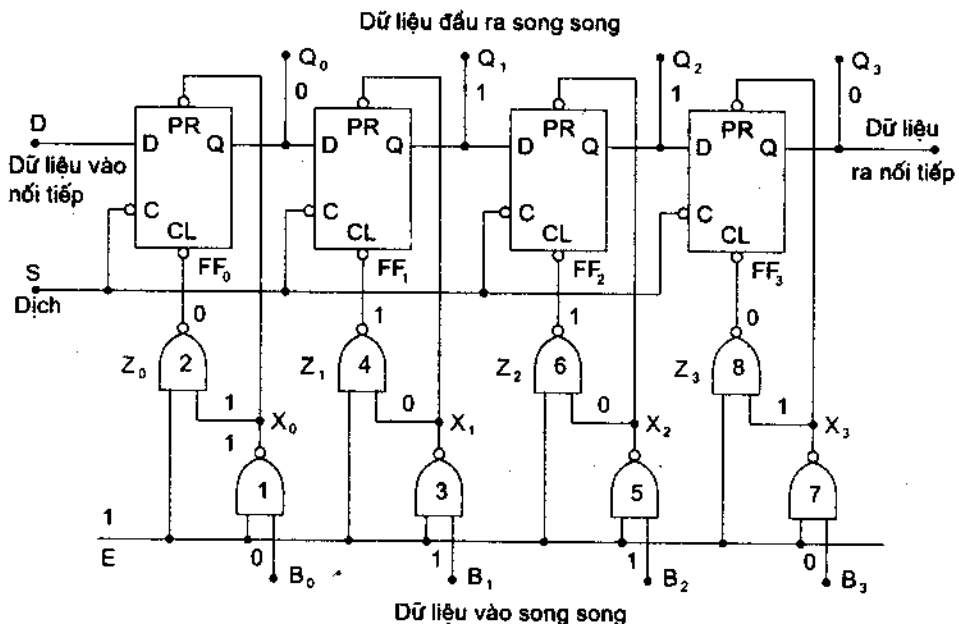
b)
Hình 4.24b. Mạch chốt dữ liệu chủ - tố đa chức năng với 2 đầu vào thiết lập PR và xóa CL cấu tạo có nguồn gốc từ DFF

- Với cổng $NAND_1$ do $E = 1$, $B_0 = 0$ nên $X_0 = 1$ và với $NAND_2$ có $Z_0 = 0$, vì đầu vào CL ở mức tích cực thấp nên $B_0 = 0$ được nhập vào FF_0 ta có $Q_0 = 0$.

- Với cổng $NAND_{3,4}$ do $E = 1$, $B_1 = 1$ nên $X_1 = 0$, $Z_1 = 1$, đầu vào PR của FF_1 ở mức tích cực thấp nên $B_1 = 1$ được nhập vào FF_1 ta nhận được $Q_1 = 1$.

Tương tự cho các đầu vào B_2 và B_3 khi đó FF_2 nhập $B_2 = 1$; $Q_2 = 1$ và FF_3 nhập $B_3 = 0$; $Q_3 = 0$, tại các đầu ra song song khi tác động đồng thời $B_0B_1B_2B_3$ ở các đầu vào song song, với điều kiện $E = 1$ (cho phép ghi nhập dữ liệu B) ta nhận được đồng thời : $Q_0Q_1Q_2Q_3 = B_0B_1B_2B_3$, đây là kiểu ghi độc vào song song ra song song của mạch.

- Trong trường hợp cần vào song song (đầu vào $B_0B_1B_2B_3$) ra nối tiếp ta tác động 3 xung S tiếp sau xung $E = 1$ để dữ liệu lần lượt xuất hiện tại đầu ra Q_3 của FF_3 theo thứ tự B_3 ra trước... B_0 ra sau cùng hay từ mã dữ liệu là $B_0B_1B_2B_3$.



Hình 4.25. Thanh ghi dịch 4 bit song song đa năng thực hiện được 4 dạng chuyển đổi dữ liệu vào - ra, nối tiếp - song song

– Trong trường hợp cần vào nối tiếp ra song song (hoặc cần vào nối tiếp ra nối tiếp) $E = 0$, sau 4 xung S (hoặc 8 xung) ta nhận được kết quả dữ liệu vào tại đầu vào nối tiếp sẽ xuất hiện tại các lối ra tương ứng.

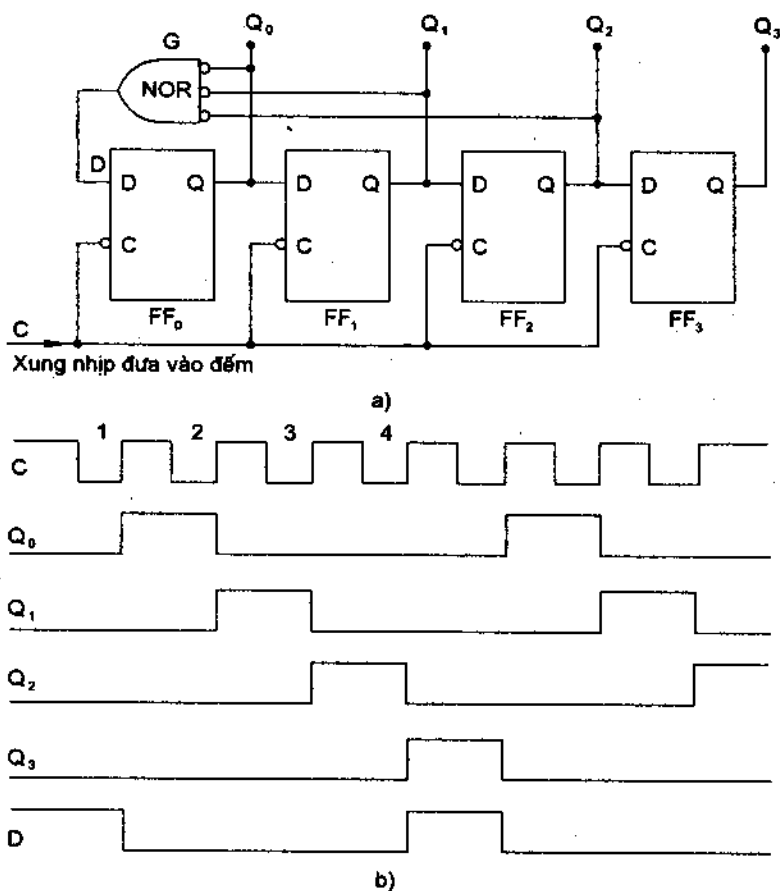
4. Bộ đếm vòng được cấu tạo từ một thanh ghi dịch có hồi tiếp, ví dụ từ đầu ra nối tiếp về lại đầu vào nối tiếp tạo thành 1 chu kỳ dịch khép kín. Nếu mạch hình 4.25 thực hiện việc nối đầu ra Q_3 vào đầu D, ban đầu chỉ cho $B_0 = 1$, $B_1 = B_2 = B_3 = 0$ và $E = 1$ FF₀ thiết lập ở $Q_0 = 1$ sau đó mỗi xung S theo nhịp tác động vào C, dữ liệu 1 từ FF₀ sẽ dịch phải dẫn đến FF₃ sau 4 nhịp nó lại quay về FF₀, thực hiện xong một chu trình đếm vòng.

Để thiết lập các trạng thái đầu ra bằng 0 sau khi xung 1 đã dịch phải và sau mỗi chu kỳ đếm tại đầu ra Q_3 ta nhận được 1 xung tín hiệu ra (ví dụ để gửi đến cấp đếm cao hơn), mạch điện hình 4.26a có chức năng thực hiện nhiệm vụ trên và tự động thực hiện công đoạn xóa bộ đếm về trạng thái ban đầu $Q_3Q_2Q_1Q_0 = 0000$. Hình 4.26b là đồ thị thời gian minh họa hoạt động của bộ đếm vòng.

Cổng NOR trong vòng hồi tiếp ở mạch điện hình 4.26a có 3 đầu vào :

$D = \overline{Q_0 + Q_1 + Q_2}$ có đặc tính tự xóa, đầu ra của NOR cho mức 0 cho tới khi đồng thời xuất hiện $Q_0 = Q_1 = Q_2 = 0$, lúc đó đầu ra lên mức cao $D = 1$ và bắt đầu 1 chu kỳ đếm mới đồng thời chỉ duy nhất lúc này có $Q_3 = 1$ tại đầu ra bộ đếm. Như vậy cứ sau 4 xung nhịp C, xung D và Q_3 bằng 1 lại xuất hiện một lần. Ta nói bộ đếm thực hiện phép chia 4 số xung đầu vào nhịp. Ta sẽ gặp lại các bộ đếm xung có cấu tạo từ các FF khác trong các tiết tiếp theo như một phần tử logic tuần tự hết sức quan trọng của kỹ thuật điện tử số.

5. Thanh ghi dịch 2 chiều: Mạch điện hình 4.26c mô tả một thanh ghi dịch 4 bit có khả năng dịch từng bit dữ liệu đã ghi trong đó đi từng vị trí sang phải hay sang trái theo lệnh điều khiển ở cổng R/L và xung dịch S.



Hình 4.26. a) Bộ đếm vòng modul 4 dùng thanh ghi dịch có hồi tiếp ;
b) Đồ thị thời gian minh họa hoạt động của bộ đếm hình 4.26a.

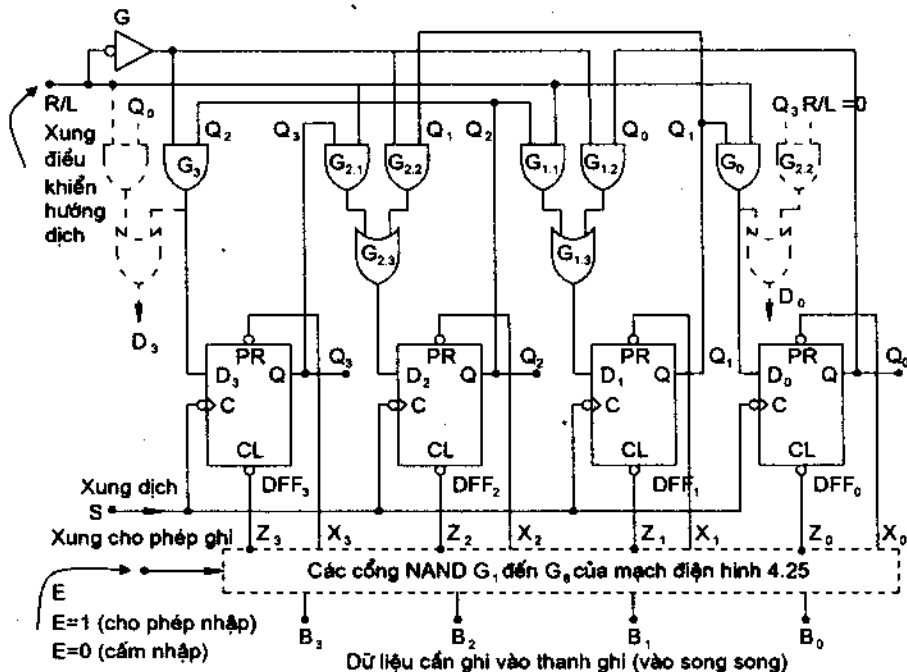
Mạch có 3 chế độ :

– Chế độ dịch sang phải khi đầu vào R/L ở mức 1, các cổng AND $G_{21}G_{11}$ và G_0 cho phép các mức ra $Q_3Q_2Q_1$ dịch sang phải nhập vào đầu vào D của FF_2, FF_1 và FF_0 tương ứng.

– Chế độ dịch trái khi R/L ở mức 0, các cổng $G_{12} G_{22}$ và G_3 mở cho phép các mức ra $Q_0 Q_1$ và Q_2 dịch trái nhập vào đầu D của $FF_1 FF_2$ và FF_3 tương ứng.

– Chế độ quay vòng được thực hiện nếu bổ sung thêm 1 cổng AND và 1 cổng OR ở mỗi đầu vào D của FF_3 và FF_0 tương tự như đối với các cổng vào

D của FF₂ và FF₁ đã có. Khi đó ở chế độ dịch phải Q₀ sẽ quay vòng nhập về FF₃ hay ở chế độ dịch trái Q₃ sẽ quay vòng nhập về FF₀ (xem đường nét đứt trên hình 4.26c).

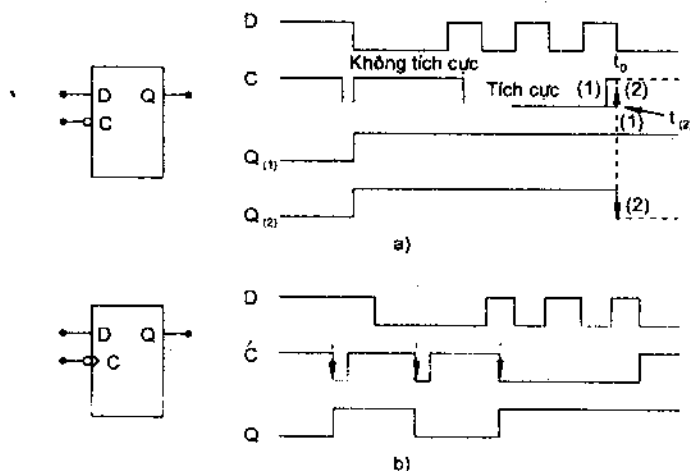


Hình 4.26c. Thanh ghi dịch 2 chiều

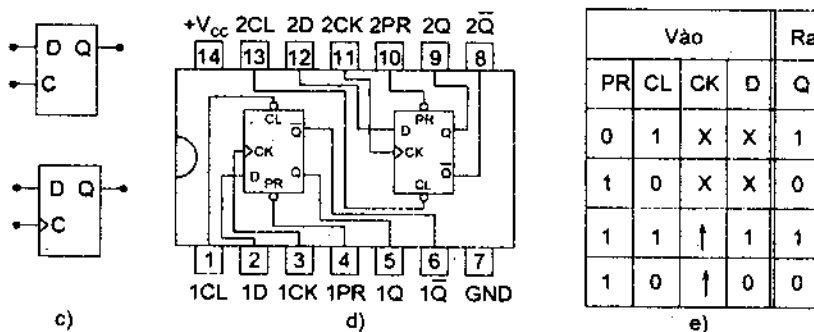
6. Phương pháp kích thích mạch chốt dữ liệu DFF hay các ô nhớ của thanh ghi dữ liệu có thể sử dụng 2 cách tùy theo đặc tính cấu tạo của mạch điện : kích bằng mức tích cực cao hay thấp và kích bằng sườn (từ thấp lên cao hay từ cao xuống thấp) khi kích thích bằng sườn xung mức đích sẽ là trạng thái mức tích cực và trong các ký hiệu quy ước cần phân biệt rõ ràng khi sử dụng. Hình 4.27a và b minh họa hai phương pháp kích này với ký hiệu quy ước phần tử nhớ kèm theo.

Trong hình 4.27a xung C ngăn không xảy ra vấn đề gì khi C chuyển xuống mức tích cực thấp để cho phép $D = 1$ nhập $Q_{(1)} = Q_{(2)} = 1$ (ở đây ta phân biệt hai trường hợp trạng thái ra Q là $Q_{(1)}$ và $Q_{(2)}$). Nếu xung C tương đối dài, khi chuyển từ mức tích cực về lại mức không tích cực vào lúc $t_{(1)}$ thì khi đó:

- Nếu $t_{(1)}$ sớm hơn t_0 lúc này xung dữ liệu đang ở mức cao ($D = 1$) vậy mức cao của D sẽ được chấp nhận ta có đồ thị $Q_{(1)} = 1$.
- Nếu $t_{(2)}$ của xung nhịp xuất hiện muộn hơn t_0 lúc này xung dữ liệu đã ở mức thấp ($D = 0$) vậy mức thấp sẽ được ghi $Q_{(2)} = 0$.



- a) Ký hiệu quy ước phần tử chốt kích bằng mức tích cực thấp và đồ thị thời gian;
b) Ký hiệu quy ước phần tử chốt kích bằng sườn đi xuống của xung nhịp và đồ thị thời gian.



Hình 4.27

- c) Ký hiệu phần tử chốt DFF kích bằng mức tích cực cao và ký hiệu phần tử kích bằng sườn đi lên.
d) Sơ đồ bố trí chân IC 7474, 74 LS74 chứa hai DFF (loại tương đương 4013 CMOS có bố trí chân khác); e) Bảng chân lý của DFF.

Đây là một nhược điểm quan trọng của phương pháp kích FF bằng mức điện thế, khi thời điểm chuyển của xung nhịp C (loại dài) không rõ ràng so với xung dữ liệu D. 1 cách ngẫu nhiên việc nhớ dữ liệu sẽ bị lỗi.

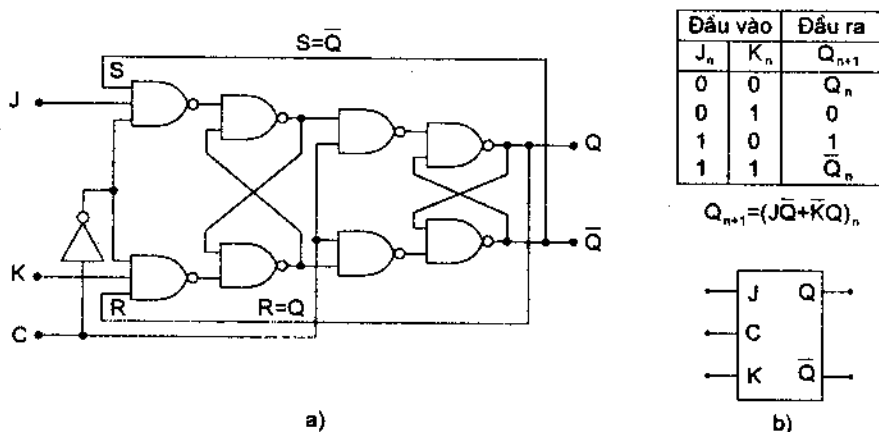
Ở trường hợp kích bằng sườn (hình 4.27b) nhược điểm này không tồn tại vì sự phản ứng của DFF làm bộ chốt nhớ dữ liệu với sườn xung nhanh hơn nhiều và mức ra Q được thiết lập gần như cùng một lúc với giá trị mức dữ liệu cần nhớ. Hình 4.27c là ký hiệu quy ước của nhóm mạch chốt kích thích bằng mức tích cực cao và kích thích bằng sườn đi lên của xung nhịp. Hình 4.27d là sơ đồ bố trí chân của IC 7474 chứa 2 DFF kèm theo bảng chân lý (hình 4.27e).

4.6. TRIGƠ VẠN NĂNG JK (JKFF)

4.6.1. Cấu tạo của JKFF

Từ Trơ chủ tớ loại RSFFT đồng bộ người ta bổ sung thêm hai đường hồi tiếp $S = \bar{Q}$ và $R = Q$ khi đó sẽ nhận được Trơ JK. Hình 4.27 là sơ đồ mạch điện của Trơ JK (viết tắt là JKFF) còn được gọi là Trơ vạn năng vì nó có khả năng biến đổi thành kiểu FF khác. Các cổng G_7 và G_8 trong mạch hình 4.21 lúc đó cần bổ sung thêm 2 đầu vào tín hiệu được đặt tên là đầu vào J (cùng cổng với đầu vào S trước đây) và đầu vào K (cùng cổng với đầu vào R trước đây). Mạch điện cấu tạo của JKFF được cho trên hình 4.28a.

Từ bảng trạng thái của JKFF thấy rõ:



Hình 4.28

- a) Mạch điện cấu tạo của JKFF dùng cổng NAND từ RSFF chủ tớ ;
b) Bảng chân lý biểu thức hàm ra và ký hiệu quy ước của JKFF chỉ có đầu vào đồng bộ.

– Các đầu vào J và K có đặc tính điều khiển đồng bộ giống như của đầu vào S và R của RSFF (3 dòng đầu của bảng chân lý, Q_{n+1} sẽ theo J nếu $J \neq K$ và $Q_{n+1} = Q_n$ nếu $J = K = 0$).

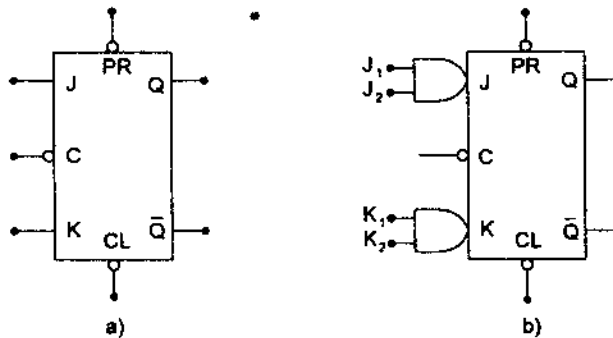
– Khi $J = K = 1$ (mức cao) mạch hoạt động như loại FFT (Trigơ đếm, xem mục 4.4.4) Q_{n+1} vẫn xác định và là trạng thái đảo của Q_n : $Q_{n+1} = \bar{Q}_n$, ta nói JKFF lật trạng thái sau mỗi xung nhịp C trong điều kiện $J = K = 1$. Trong khi đó, với tổ hợp cấm này RSFF không được phép làm việc.

4.6.2. Mở rộng chức năng của JKFF

Để mở rộng chức năng hoạt động của JKFF, cũng như DFF, thực hiện cải tiến mạch hình 4.28a như sau:

– Bổ sung 2 đầu vào điều khiển trực tiếp PR (thiết lập mức $Q = 1$) và CL (thiết lập mức $Q = 0$ hay gọi là xóa về 0) có mức tích cực thấp. Các đầu vào này ở vị trí tương tự hình 4.24b.

– Bổ sung các cổng AND ở đầu vào đồng bộ để tạo ra cổng nhiều đầu vào J và đầu vào K như thể hiện trên hình 4.29.



Hình 4.29. Ký hiệu JKFF có đầu vào PR, CL và C mức tích cực thấp
a) JKFF đơn ; b) JKFF nhiều đầu vào J và K

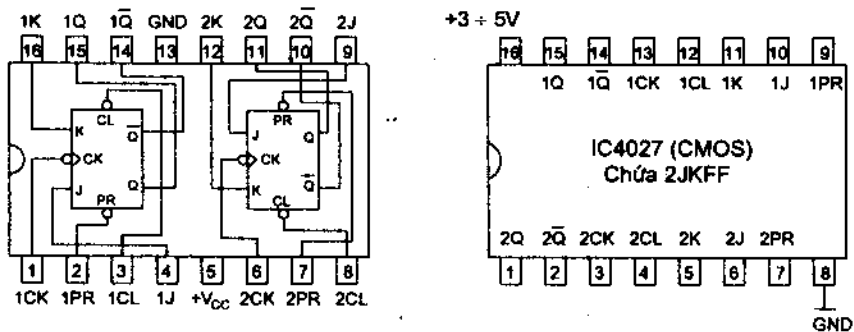
4.6.3. Hình 4.30 cho sơ đồ bố trí chân của IC 7476 / 74 LS76 loại TTL hay 4027 loại CMOS có chứa 2 JKFF trong 1 vỏ.

1. Bảng chân lý 4.30c của JKFF chứa đầy đủ thông tin cho biết cách sử dụng đúng các đầu vào để đạt tới một trạng thái ra mong muốn.

– Khi có xung $PR = 0$ (mức tích cực thấp) sẽ thiết lập trực tiếp tại đầu ra $Q = 1$, $\bar{Q} = 0$. Khi có xung $CL = 0$ xóa trực tiếp $Q = 0$, $\bar{Q} = 1$.

– Khi $PR = CL = 1$ sử dụng các đầu vào đồng bộ với xung nhịp, xung CK kích thích bằng mức tích cực thấp ($\bar{CK}$) với loại IC 7476 còn với loại 74LS76 kích bằng sườn đi xuống của xung CK. Lúc đó nếu $J \neq K$ thì Q theo trị của J, $\bar{Q}$ theo trị của K; nếu $J = K = 1$, Q và $\bar{Q}$ lật trạng thái (đổi trị cho nhau).

Chế độ này gọi là chế độ bập bênh (toggle) mà ở các FF khác không có, vì lý do này JKFF còn được gọi là FF bập bênh.



Hình 4.30 a) Sơ đồ bố trí chân của IC 74LS76 chứa hai JKFF (với IC loại 7476 kích bằng mức thấp) ;
b) Sơ đồ bố trí chân của IC 4027 CMOS chứa hai JKFF (giống 74LS76)

c) Bảng chân lý của IC 7476 (74LS76) khi có 2 đầu vào điều khiển trực tiếp (chú ý xung CK: Với 7476 mức tích cực thấp áp tới. Với 74LS76 sườn đi xuống của xung nhịp sẽ kích JKFF)

Đầu vào					Đầu ra	
PR	CLR	CK	J	K	Q	Q̄
0	1	x	x	x	1	0
1	0	x	x	x	0	1
1	1		1	0	1	0
1	1		0	1	0	1
1	1		1	1	Lật bập bênh	

2. Hình 4.31a và b giải thích hai phương thức hoạt động của JKFF bằng đồ thị thời gian minh họa, đồ thị chuyển của Q hoàn toàn tuân thủ bảng 4.30c đã có.

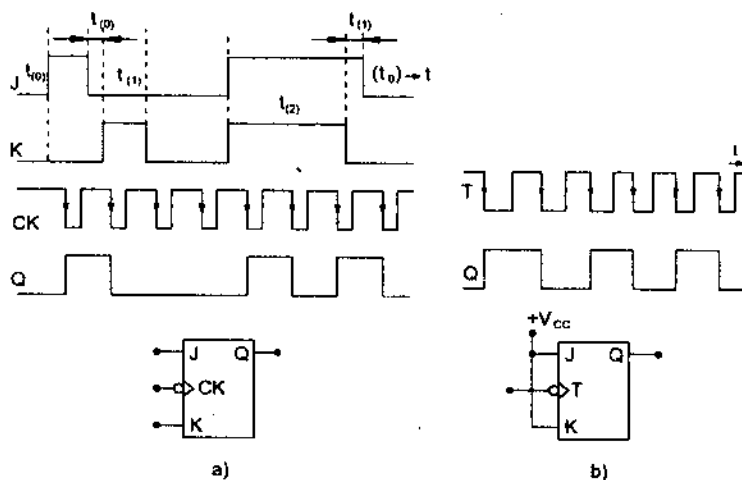
– Trên đồ thị hình 4.31a có 3 khoảng thời gian cần chú ý:

$t_{(0)}$ là các khoảng $J = K = 0$ do đó Q giữ nguyên trạng thái cũ mới khi có sườn xuống của xung nhịp (không đổi trạng thái).

$t_{(1)}$ là các khoảng thời gian ở đó $J \neq K$ và Q sẽ có giá trị theo giá trị của J (trong ví dụ lúc có nhịp $J = 0$ thì $Q = 0$)

$t_{(2)}$ là khoảng thời gian ở đó $J = K = 1$ lúc đó Q sẽ lật trạng thái mỗi khi có sườn đi xuống của xung CK xuất hiện.

– Đồ thị hình 4.31b JKFF sẽ đổi trạng thái mỗi khi có sườn đi xuống của CK- ta gọi đây là chế độ đếm xung nhịp của JKFF lúc đó còn gọi là kiểu TFF (Trigơ đếm). Điều kiện xảy ra chế độ này là $J = K = 1$ (cả hai đầu vào này được nối tới $+V_{cc}$).

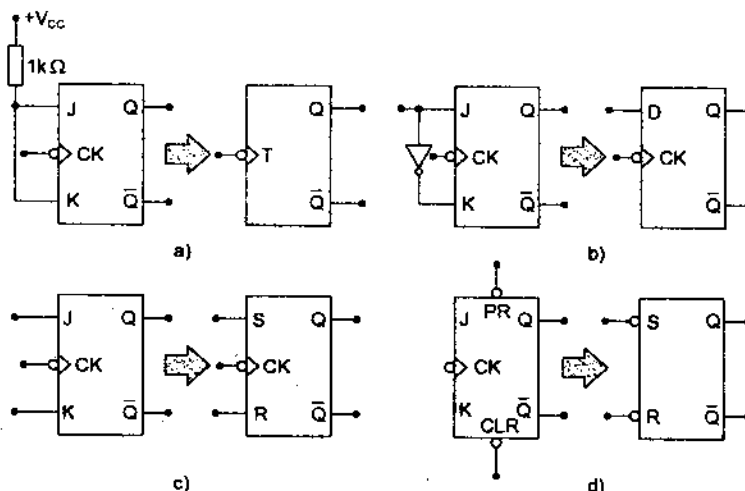


Hình 4.31

- a) Trạng thái ra Q của JKFF kích bằng sườn đi xuống của xung nhịp CK (dữ liệu nhập vào và xuất hiện ở đầu ra tại sườn xuống của CK).
b) Chế độ bập bênh của JKFF có được khi $J = K = 1$ (Q lật sau mỗi sườn đi xuống của xung CK = T).

4.6.4. Dùng JKFF thay cho mạch chốt dữ liệu và thay RSFF

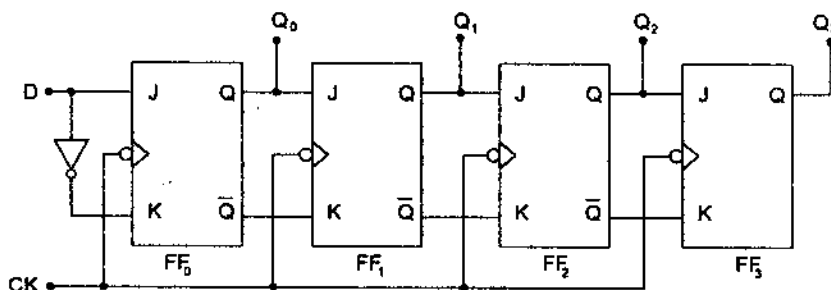
I. Hình 4.32 minh họa 4 cách sử dụng JKFF thay cho RSFF, DFF và TFF chứng minh tính chất đa năng của JKFF.



Hình 4.32. Các cách thay thế FF khác bằng JKFF vạn năng

- a) Cho $J = K = 1$ thường xuyên để JKFF ở chế độ bập bênh có TFF (Trigơ đếm) hoặc để thả nổi J và K;
b) Cho $\bar{J} = K$ nhận được Trigơ trễ DFF (mạch chốt dữ liệu 1 bit xem phần 4.3);
c) Sử dụng trực tiếp JKFF như một RSFF chấp nhận làm việc cả khi $J = K = 1$, đầu vào J coi như S đầu vào K coi như R;
d) Sử dụng 2 đầu vào điều khiển trực tiếp PR (thiết lập) như S, CLR (xóa) như R tích cực đều ở mức thấp, bỏ các đầu vào đồng bộ J, K, CK có RSFF không đồng bộ (như mạch hình 4.1).

2. Mạch điện hình 4.32e thể hiện một thanh ghi dịch nối tiếp trong đó các RSFF hay DFF chủ tố đã được thay thế bằng JKFF sau khi đã biến đổi mạch điện bằng cách như hình 4.32b đã thể hiện đối với FF_0 , các ô tiếp theo của thanh ghi chỉ cần cho $J = Q$ và $K = \bar{Q}$, khi ghép nối sẽ cho phép cấm trạng thái bập bênh $J = K = 1$ với mọi ô nhớ của thanh ghi.



Hình 4.32e. Thanh ghi dịch nối tiếp 4 bit dùng JKFF cải tiến thành DFF

4.7. BỘ ĐẾM

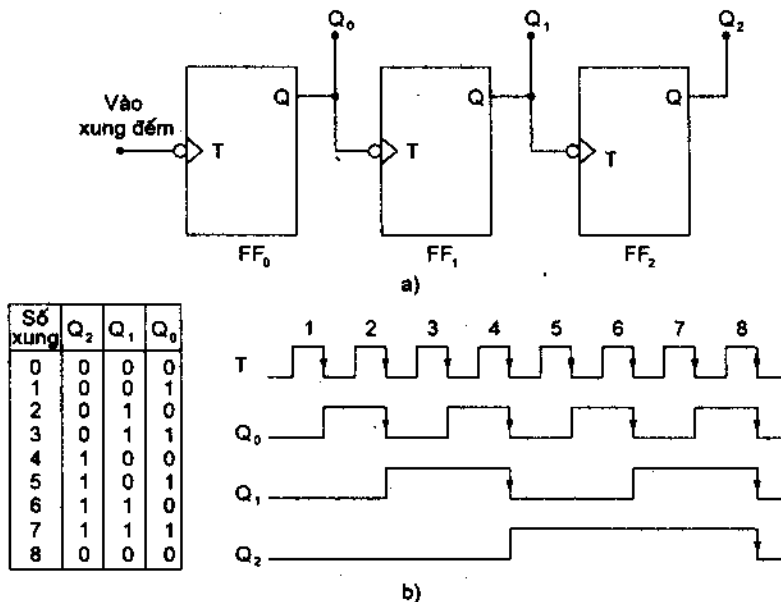
4.7.1. Bộ đếm xung tuần tự

Thực hiện ghép liên tiếp các TFF hoạt động bập bênh của các Trigo đếm mang tính dây chuyền ta sẽ nhận được bộ đếm xung kiểu tuần tự : Cứ mỗi xung nhịp áp đến đầu vào T sẽ được bộ đếm cộng lại trong hệ nhị phân, tổng số xung vào sẽ xuất hiện ở dạng nhị phân tại các đầu ra của TFF. Tổng quát khi ghép n Trigo đếm liên tiếp nhau số trạng thái nhị phân tương ứng với số xung tối đa có thể đếm được là $2^n - 1$ gọi là dung lượng (sức chứa) của bộ đếm. Bộ đếm xung mô tả trên đây gọi là bộ đếm nhị phân thực chất là phép chia hai liên tiếp số lượng xung nhịp ở đầu vào T.

Hình 4.33a và 4.34a là cấu trúc mạch điện của hai dạng bộ đếm nhị phân phổ biến nhất dùng TFF (hay JKFF ở chế độ $J = K = 1$) gọi tên là bộ đếm nhị phân kiểu cộng xung nối tiếp (hình 4.33a) và kiểu trừ xung (hình 4.34a) với số xung đếm tối đa là $2^3 - 1 = 7$ xung, chính là phép đếm trong hệ đếm 8. Ta có bảng trạng thái, đồ thị thời gian minh họa hoạt động của các mạch 4.33a và 4.34a thể hiện trên hình 4.33b và 4.34b.

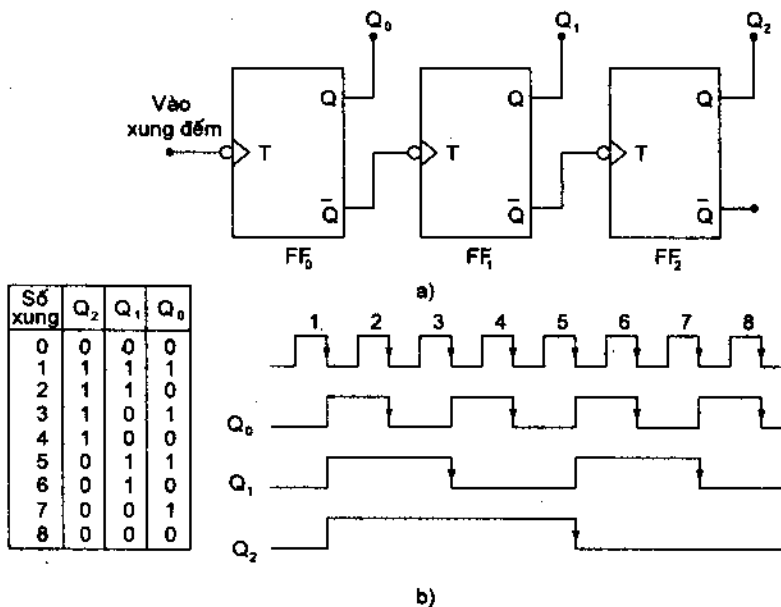
Có hai nhận xét quan trọng từ bảng trạng thái của bộ đếm nhị phân :

- FF phía sau sẽ lật trạng thái khi FF phía trước chuyển từ mức cao về mức thấp (sườn đi xuống của xung).
- FF phía sau sẽ chuyển khi mọi FF phía trước đã ở mức 1. Nhận xét đầu giúp ta giải thích kết cấu và hoạt động của các mạch 4.33a và 4.34a : cứ sau mỗi sườn đi xuống của xung đếm (từ mức 1 về mức 0) FF_0 sẽ chuyển giá trị,



Hình 4.33. a) Mạch đếm hệ 2 kích thích không đồng bộ bằng sườn đi xuống dùng TFF (cộng xung-đếm lên)

b) Bảng chân lý và đồ thị minh họa hoạt động của mạch đếm hình 4.33a cộng dồn xung.



Hình 4.34

a) Mạch đếm hệ hai kích thích không đồng bộ bằng sườn đi xuống (đếm xuống trừ xung)

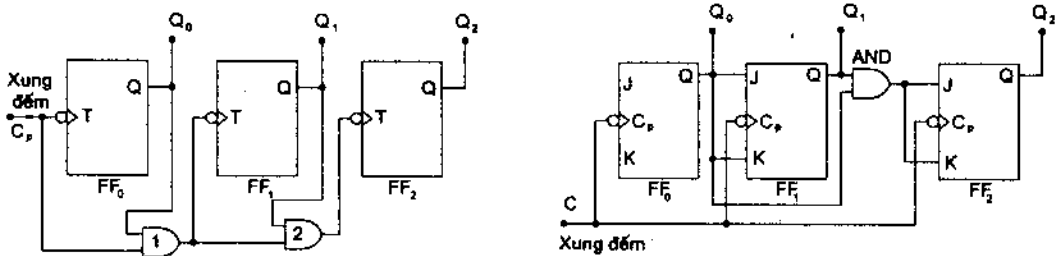
b) Bảng chân lý và đồ thị thời gian minh họa hoạt động của mạch đếm trừ dẫn số xung.

FF_1 và FF_2 cùng có thuộc tính tương tự khi Q_0 và Q_1 chuyển từ 1 về 0. Ở mạch (4.34a) đếm xuống thực hiện bằng cách trừ dần số xung, sau lần kích thích thứ nhất có $Q_0 = Q_1 = Q_2 = 1$, vì các đầu ra $\bar{Q}$ được nối với đầu vào T của FF phía sau nên khi Q từ 0 lên 1 thì $\bar{Q}$ từ 1 xuống 0 sẽ kích thích chuyển cho FF sau như đồ thị thời gian hình 4.34b đã chỉ rõ.

Khi sử dụng nhiều TFF mắc liên tiếp, dung lượng của bộ đếm sẽ tăng, ví dụ với 5 TFF sẽ đếm được $2^5 - 1 = 31$ xung đến xung thứ 32 thì $Q_4 = Q_3 = Q_2 = Q_1 = Q_0 = 0$ (về trạng thái ban đầu). Tuy nhiên việc ghép nhiều sẽ tích lũy sự chậm trễ của mỗi FF lại, đây là nhược điểm của mạch đếm kích thích không đồng bộ dạng hình 4.33 và 4.34 còn gọi là kiểu đếm nối tiếp.

4.7.2. Bộ đếm 2 kích thích đồng bộ

Xuất phát từ nhận xét thứ hai đối với bảng chân lý bộ đếm lên (hình 4.33a) là FF phía sau chỉ chuyển khi tất cả các FF phía trước nó đã ở 1, ta xây dựng được bộ đếm 2 kích thích đồng bộ (còn gọi là bộ đếm kiểu song song) như trên các hình 4.35 và 4.36.

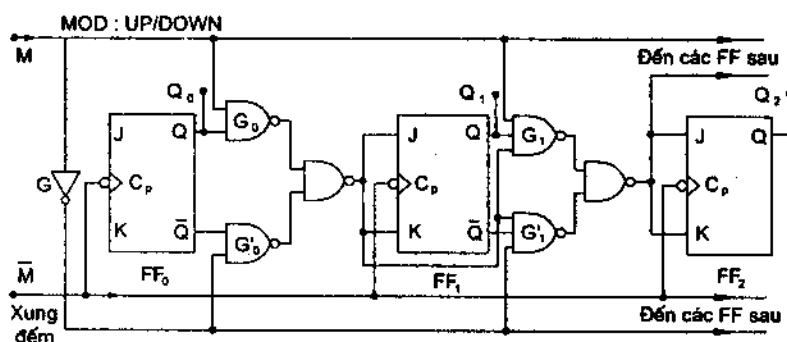


Hình 4.35. Bộ đếm 2 đồng bộ dùng TFF (3bít) ; **Hình 4.36.** Bộ đếm 2 đồng bộ dùng JKFF (3bít)

Ở mạch hình 4.35, mỗi khi xung đếm vào FF_0 lật theo sườn âm của xung C_p cổng AND 1 chọn điều kiện $C_p = Q_0 = 1$ để cho phép FF_1 chuyển, cổng AND 2 chọn điều kiện $Q_0 = Q_1 = C_p = 1$ để cho phép FF_2 chuyển, sau 7 xung đếm C_p , trạng thái của FF_0 , FF_1 và FF_2 tại các đầu ra song song là 111 (bộ đếm 3 bít đã đầy). Mạch hình 4.36 sử dụng JKFF chế độ bập bênh để thực hiện việc đếm xung, ở đây sử dụng các đầu vào JK để điều khiển các FF_1 và FF_2 tương ứng lúc mỗi xung đếm C_p áp tới cửa vào nhịp của chúng. FF_0 chuyển sau mỗi sườn đi xuống của C, FF_1 chuyển khi $C_p = Q_0 = J = K = 1$. FF_2 chuyển khi $C_p = 1$ và $Q_0 = Q_1 = J = K = 1$, tất cả các chuyển của FF_1 và FF_2 đều xảy ra ở sườn đi xuống của các xung kích thích. Có thể thay thế cổng AND và FF_2 bằng một JKFF có 2 đầu vào $J_1 J_2$ và 2 đầu vào $K_1 K_2$ như hình 4.29b.

4.7.3. Bộ đếm lên xuống kích thích đồng bộ

Hình 4.37 giới thiệu mạch điện của bộ đếm xung lên xuống đồng bộ. Ở mạch này có một đầu vào chọn M, khi $M = 1$ các cổng G_0 và G_1 được chọn cho chế độ đếm lên, khi các Q_0, Q_1 ở mức 1 sẽ cho phép các FF_1 và FF_2 chuyển như bộ đếm đã xét ở hình 4.36. Còn khi $M = 0$ các cổng G_0, G_1 sẽ cấm Q và qua bộ đảo G có $\overline{M} = 1$ chọn G_0', G_1' đến khi $\overline{Q}_0, \overline{Q}_1$ ở mức 1 sẽ điều khiển JK của các FF tiếp sau cho phép nó chuyển với phương thức làm việc như mạch hình 4.34a chế độ đếm xuống.



Hình 4.37. Bộ đếm 2 chiều lên xuống (3bit) đồng bộ

4.7.4. Bộ đếm BCD 8421 có cấu tạo gồm 4 FF (loại T hay loại JK chủ chốt) có các trạng thái ra $Q_3Q_2Q_1Q_0 = DCBA$ của FF_D , FF_C , FF_B và FF_A biểu diễn một số thập phân (là số xung đếm được) dưới dạng 1 từ mã BCD 8421, tức là trạng thái của $Q_3 = D$ ứng với trọng số 2^3 ..., của $Q_0 = A$ ứng với trọng số 2^0 . Bảng chân lý cho bộ đếm BCD 8421 được cho trên hình 4.38a. Mạch điện thực hiện với hai dạng kích thích không đồng bộ cho trên hình 4.38b và kích thích đồng bộ trên hình 4.38 d.

I. Quan sát bảng chân lý 4.38a ta thấy:

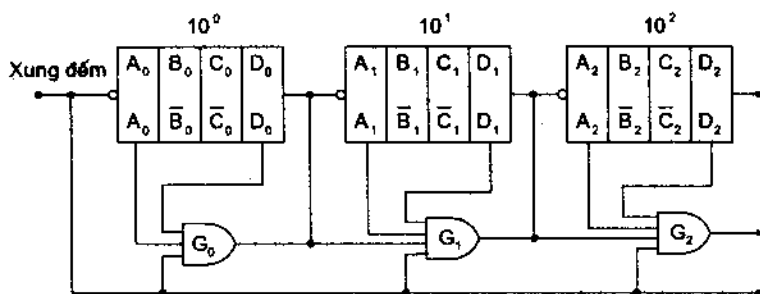
- FF_A chuyển đổi trạng thái sau mỗi lần kích thích của xung đếm.
- FF_B chỉ chuyển đổi trạng thái khi FF_A từ 1 xuống 0 trừ trường hợp cuối cùng lúc FF_D ở mức cao ($D = 1$).
- FF_C chuyển đổi trạng thái theo đầu ra B của FF_B khi B từ 1 xuống 0.
- FF_D coi như chuyển đổi giống FF_A tức là theo trị A từ 1 xuống 0 nhưng bị cấm trong suốt thời gian có xung nhịp từ 0 tới 7, cho tới khi $B = C = 1$ mới chuyển.

quay về FF_B thông qua đầu vào $J = K = \bar{D} = 0$ của nó, dẫn tới $B = 0$ kéo theo $C = 0$. Theo giản đồ thời gian hình 4.38c khi hết xung thứ 8 trạng thái ra của bộ đếm là 1000 tương ứng với việc phân tích vừa xong. Đến xung thứ 9 mạch làm việc bình thường chỉ có FF_A chuyển nên DCBA = 1001. Khi xung thứ 10 xuất hiện (xét sườn đi xuống) $A = C = 0$ theo trật tự bình thường của bộ đếm nhị phân, do đường hồi tiếp $\bar{Q} = \bar{D} = 0$ nên với FF_B có $A = C_p = J = K = 0$ kết quả $B = 0$ qua AND làm J của FF_D xuống mức 0 và $D = 0$. Hai giá trị B và D theo trật tự bộ đếm nhị phân đáng lẽ ở 1 nhưng do mạch hồi tiếp, sau xung thứ 10 ta nhận được DCBA = 0000 kết thúc 1 chu trình đếm của bộ đếm BCD 8421.

3. Với mạch điện hình 4.38d (cấu trúc song song) việc kích thích các FF thực hiện đồng bộ theo từng nhịp của xung đếm C_p , trừ FF_A làm việc ở chế độ bập bênh thay đổi trạng thái theo mỗi nhịp, còn FF_B, FF_C và FF_D chịu sự điều khiển cho phép hay không cho phép chuyển thông qua các đầu vào điều khiển J và K của chúng, tín hiệu điều khiển này do các cổng logic $G_1 = A\bar{D} = J_B$; $G_2 = A.B = J_C$; $G_3 = A.B = K_C$ và $G_4 = ABC = J_D$ tạo ra (lưu ý $K_B = K_D = A$, J_A , K_A thả nổi hay nối với +5V qua điện trở 1kΩ tương tự như các đầu JK của FF_A, FF_C, đầu K của FF_D trong hình 4.38b).

Các hàm logic G_1, G_2, G_3, G_4 có được nhờ các nhận xét từ bảng chân lý 4.38a sau đó tối thiểu hoá chúng bằng bảng các nô.

4. Để nhận được bộ đếm BCD nhiều decac cần ghép nối các bộ đếm 4 bit hình 4.38b hay 4.38d theo kiểu nối tiếp (không đồng bộ) hay kiểu song song (đồng bộ). Hình 4.39 mô tả cách nối 3 bộ đếm 4 bit BCD để tạo ra được bộ đếm hàng trăm xung kiểu kích thích đồng bộ, các cổng AND G_0, G_1 và G_2 có nhiệm vụ cho phép các cấp hàng chục hàng trăm hoặc hàng nghìn được phép đếm khi bộ đếm BCD ở cấp dưới kể nó đã đạt tới 9 xung (1001).



Hình 4.39. Bộ đếm BCD tới hàng trăm

4.7.5. Bộ đếm số xung theo một hệ đếm bất kỳ

1. Khi ghép các TFF hay JKFF lại, mỗi FF thực hiện một bộ đếm cơ số 2 1 bit vậy các bộ đếm cơ số 2^n có được nhờ ghép liên tiếp n bộ đếm 2 1 bit lại với nhau kiểu nối tiếp (không đồng bộ) hay kiểu song song (đồng bộ). Ví dụ mạch điện hình 4.33a thực hiện đếm trong hệ 8, nếu bớt đi 1 ô FF₂ nó sẽ thực hiện đếm các số 0, 1, 2, 3, 0, 1... tức là đếm trong hệ $2^2 = 4$, hoặc nếu thêm vào 1 ô FF₃, nó sẽ thực hiện đếm xung trong hệ $16 = 2^4$...

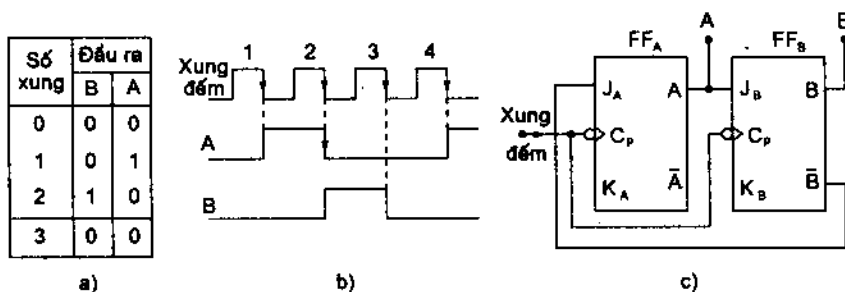
2. Bằng phương pháp sử dụng mạch hồi tiếp thích hợp, có thể tạo ra một bộ đếm để thực hiện việc đếm xung trong hệ đếm N bất kỳ với số xung đếm là 0, 1, 2, ..., N - 1 (N là số nguyên dương). Các hình 4.40, 4.41 là mạch điện thực hiện đếm xung trong hệ đếm 3, 5 tương ứng. Khi sử dụng số lượng ô đếm n FF phải tuân theo điều kiện $2^n \geq N$.

a) Khi chọn N = 3 (hệ đếm 3), cần dùng 2 ô đếm nhị phân, như vậy tùy theo cách lựa chọn xuất hiện 4 dạng mạch đếm hệ 3 là :

- Dùng các số 0, 1, 2 (bỏ 3) thường được sử dụng
- Dùng các số 1, 2, 3 (bỏ 0)
- Dùng các số 0, 2, 3 (bỏ 1)
- Dùng các số 0, 1, 3 (bỏ 2)

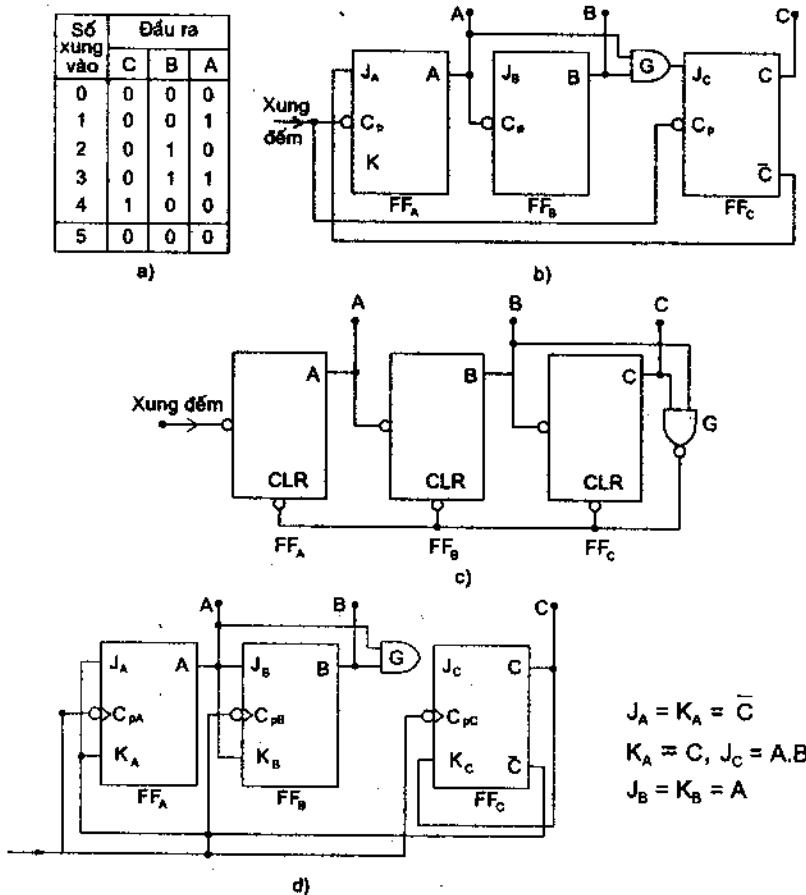
Ví dụ : Chọn các số đếm 0, 1, 2, bảng chân lý, giản đồ thời gian và mạch điện thực hiện cho trên hình 4.40.

Từ bảng chân lý 4.40a mạch hoạt động như một bộ đếm hệ 2 cho đến hết xung thứ 2 (trạng thái B = 1, A = 0) sau đó khi có xung thứ 3, FF_A bị khóa giữ ở 0 kéo theo FF_B ở 0 vì A = J_B = 0, trong mạch có một đường hồi tiếp $\bar{B} = J_A$ để khi B = 1 thì $\bar{B} = J_A = 0$ FF_A do điều kiện này bị khóa lại ở 0 (A = 0) khi xung thứ 3 áp tới đầu vào C_p.



Hình 4.40. Bộ đếm hệ 3. a) Bảng chân lý ;
b) Đồ thị thời gian minh họa ; c) Mạch điện bộ đếm hệ 3.

b) Khi chọn $N = 5$ ta cần dùng 3 ô đếm nhị phân và có nhiều cách thực hiện, thường gặp nhất là sử dụng các số tự nhiên thập phân đầu tiên: 0, 1, 2, 3, 4, 0, 1... Bảng chân lý và mạch điện của một dạng bộ đếm hệ 5 cho trên hình 4.41a và b:



Hình 4.41. a) Bảng chân lý bộ đếm hệ 5

Một dạng mạch bộ đếm hệ 5 không đồng bộ (b) và đồng bộ (d) dùng hồi tiếp loại bỏ các trạng thái không sử dụng.

c) Mạch đếm hệ 5 dùng phương pháp loại bỏ trực tiếp các trạng thái không sử dụng.

– Từ bảng chân lý (4.41a) có nhận xét:

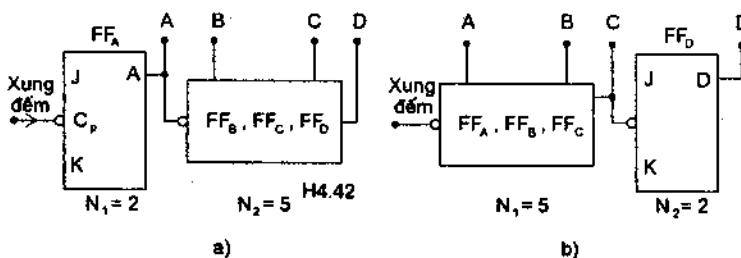
+ FF_A và FF_B đếm tự nhiên đến 3 xung trong khi trạng thái ra của FF_C có giá trị $C = 0$.

+ Đến xung đếm thứ 4, FF_C chuyển $C = 1$, khi đó $A = B = 0$. Từ đó trở đi FF_A và FF_B bị giữ lại ở trạng thái $A = B = 0$ cho đến khi xung thứ 5 xuất hiện và C tự trở về 0.

- Mạch điện hình 4.41b thực hiện một vòng hồi tiếp $J_A = \overline{C}$, minh họa rõ hoạt động của bộ đếm hệ 5 như sau : khi hết xung thứ 4 ($CBA = 100$) theo trật tự nhị phân đếm bình thường $C = 1$, $\overline{C} = J_A = 0$ nên xung thứ 5 không làm chuyển FF_A , lối ra có $A = 0$ và vì vậy FF_B giữ nguyên trạng thái trước là $B = 0$ qua cổng AND $G = A \cdot B = J_C = 0$ điều khiển FF_C trở về 0, kết quả là xung thứ 5 gây ra trạng thái $CBA = 000$ kết thúc một chu kỳ của bộ đếm 5.

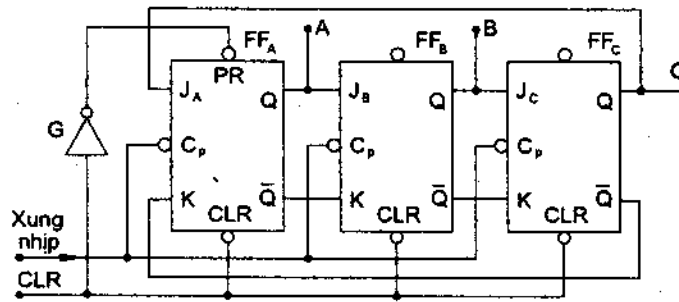
- Mạch điện hình 4.44c dùng các đầu vào điều khiển trực tiếp CLR để loại bỏ các trạng thái không sử dụng. Mạch đếm tự nhiên tới trạng thái $N - 1 = 5 - 1 = 4$, sau đó khi bị kích thích bởi xung thứ 5 cổng G (NAND) phát hiện trạng thái này và xóa toàn bộ các FF_A , FF_B và FF_C về 0 (trạng thái xung thứ 5) là $CBA = 101$, do đó $G = \overline{AC} = 0$ đưa về các đầu vào xóa điều khiển trực tiếp $C = B = A = 0$ (mức $C = 1$ có thể dùng để đưa tới 1 bộ đếm cấp cao hơn tiếp theo của bộ đếm 5).

3. Có thể thực hiện 1 bộ đếm hệ N bằng cách ghép liên tiếp hai bộ đếm hệ N_1 và N_2 sao cho $N_1 \cdot N_2 = N$. Ví dụ với $N_1 = 2$, $N_2 = 5$ ta sẽ thực hiện được bộ đếm hệ $N = N_1 \cdot N_2 = 10$. Như vậy có hai cách thực hiện như trên hình 4.42. Mạch hình 4.42a cho từ mã DCBA dạng tuần tự trong biểu diễn BCD 8421, dạng hình 4.42b sẽ cho loại mã DCBA đặc biệt với nhóm tín hiệu ra có hai nửa chu kỳ hoàn toàn giống nhau (chia đôi tần số xung ra của khối ABC) trong khi dạng (4.42a) thực hiện việc chia 8 dãy xung xuất hiện ở A.



Hình 4.42. Hai cách thực hiện bộ đếm hệ 10: 2 x 5 hay 5 x 2
a) Dạng 5 x 2; b) Dạng 2 x 5

4. Có thể hiểu một cách tổng quát một bộ đếm xung trong hệ đếm N . (đếm từ 0 đến $N - 1$ xung) là một bộ chia tần số của xung đếm đi N lần. Ví dụ, với $N = 3, 5, 10$ đã xét ta đã có các bộ chia tần số xung nhịp C_p đi 3, 5, 10 lần tương ứng ; đồng thời có thể thực hiện phép chia 3.10, 3.5, 5.10, hay 3.5.10 nhờ ghép liên tiếp các bộ chia này với nhau một cách thích hợp.



Hình 4.43. Bộ đếm vòng dung lượng 3

4.7.6. Bộ đếm vòng

Trong phần xét MSFF với các ứng dụng ở 4.5.3 ta đã giới thiệu một bộ đếm vòng dùng thanh ghi dịch dạng DFF chủ tớ. Trong phần này, khi sử dụng các JKFF thực hiện việc đếm vòng kết cấu bộ đếm trở nên đơn giản hơn nhiều.

1. Mạch đếm vòng thuận (Ring Counter) dùng JKFF cho trên hình 4.43 (còn gọi là mạch ghi tuần hoàn).

– Ban đầu cho một xung xóa $CLR = 0$ trạng thái ban đầu $C = B = A = 0$ được thiết lập. Tiếp sau cho một xung $CLR = 1$ qua cổng đảo G tới đầu vào PR của FF_A nên mức $A = 1$ được thiết lập, ta có trạng thái thứ nhất của bộ đếm là $ABC = 100$, khi xung nhịp xuất hiện ở các đầu vào C_p chỉ có FF_B thỏa mãn $J_B = A = 1$ nên $B = 1$, vì $C = 0 = J_A$ nên FF_A không phản ứng với xung C_p này, ta có trạng thái tiếp của bộ đếm là $ABC = 010$, khi xung nhịp tiếp theo xuất hiện chỉ $J_C = B = 1$ nên FF_C được kích thích, và $C = 1$ ta có trạng thái tiếp là $ABC = 001$, xung C_p tiếp theo làm FF_A chuyển A lên 1, trạng thái 100 lại xuất hiện... Vậy cứ 3 xung nhịp mạch thực hiện được 1 vòng đếm với giá trị 1 được dịch từ trái sang phải 1 cách tuần hoàn (bộ đếm dung lượng 3). Nếu sử dụng n.FF ta được bộ đếm vòng với dung lượng là n.

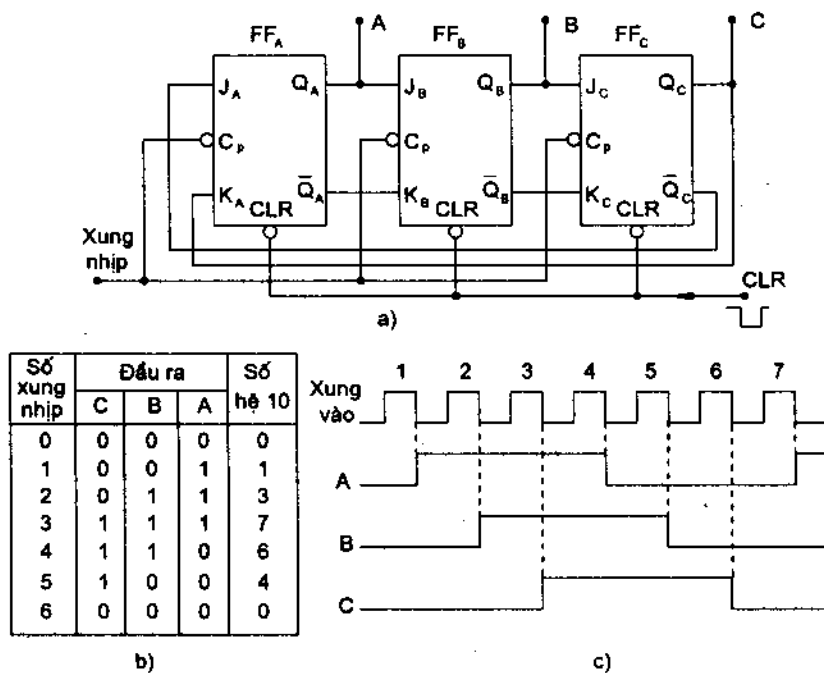
– Dạng sóng ra của mạch đếm vòng đặc biệt chỉ có một xung đơn lẻ tại đầu ra của một trong các FF được kích hoạt tích cực đồng bộ với xung nhịp.

– Bằng cách kích thích trước vào các cổng PR của một vài FF trong mạch vòng (mức tích cực thấp) có thể tạo ra nhiều dạng sóng đặc biệt tại các đầu ra song song của các FF trong vòng kín này sau nhịp thứ k nào đó.

2. Mạch đếm vòng trên hình 4.44 có được nhờ thay đổi vòng hồi tiếp:
 $Q_C = K_A$ và $\bar{Q}_C = J_A$.

– Ban đầu cho xung $CLR = 0$ (xóa các FF về 0, $A = B = C = 0$)

Khi xung đếm thứ nhất xuất hiện, FF_A được kích thích (vì $\bar{Q}_C = \bar{C} = 1 = J_A$ và $Q_C = C = K_A = 0, A = 1$) trạng thái 100 = ABC tự động được xác lập, xung thứ 2 kích thích FF_B làm $B = 1$ ($J_B = Q_A = 1, K_B = \bar{Q}_A = 0$) A vẫn ở 1 vì đảm bảo điều kiện $J_A = 1, K_A = 0, C_p = 1$, ta có ABC = 110. Xung thứ 3 tiếp tục duy trì A = 1, do $J_C = B = 1, K_C = \bar{B} = 0, C_p = 1$ nên C = 1, ta có trạng thái thứ 3 của bộ đếm là ABC = 111. Xung thứ 4 làm FF_A lật: A = 0 ($J_A = \bar{C} = 0, K_A = C = 1, C_p = 1$), trạng thái thứ 4 là ABC = 011. Khi xung thứ 5 xuất hiện chỉ có FF_B được kích thích, ABC = 001. Đến xung thứ 6 chỉ có FF_C được kích thích ta có ABC = 000. Bộ đếm hoàn thành 1 vòng đếm (đã bỏ qua 2 trạng thái 010 và 101).



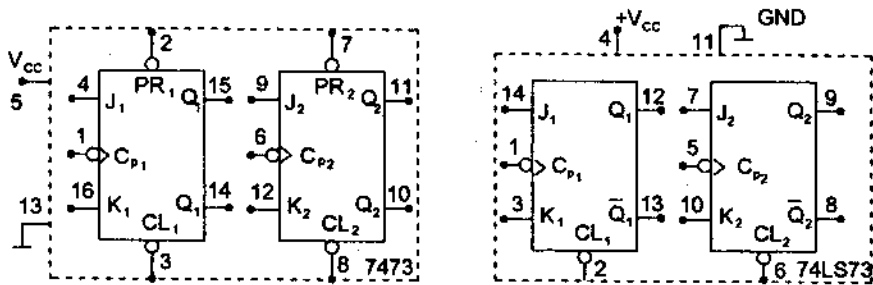
Hình 4.44. a) Bộ đếm vòng nghịch dụng lượng 6
b) Bảng trạng thái bộ đếm vòng nghịch hình 4.44a
c) Đồ thị thời gian minh họa hoạt động theo xung nhíp

4.7.7. Bộ đếm có giá trị ban đầu được thiết lập trước

1. Khi cần cho bộ đếm thực hiện đếm xung bắt đầu từ một trạng thái bất kỳ nào đó trong chu trình đếm của nó trở đi, có thể thực hiện việc thiết lập

4.8. MỘT SỐ IC DFF, JKFF, TFF, THANH GHI DỊCH VÀ BỘ ĐẾM THÔNG DỤNG

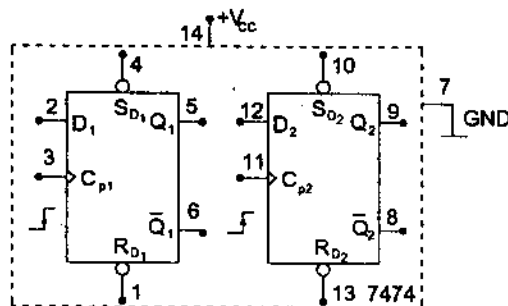
Có rất nhiều vi điện tử thực hiện các chức năng Flip Flop, thanh ghi dịch, bộ đếm thuộc dòng công nghệ TTL hay CMOS. Phần này sẽ giới thiệu tóm tắt một số IC thông dụng nhất về chức năng, bố trí chân và các vấn đề điều khiển khi sử dụng chúng.



Hình 4.46

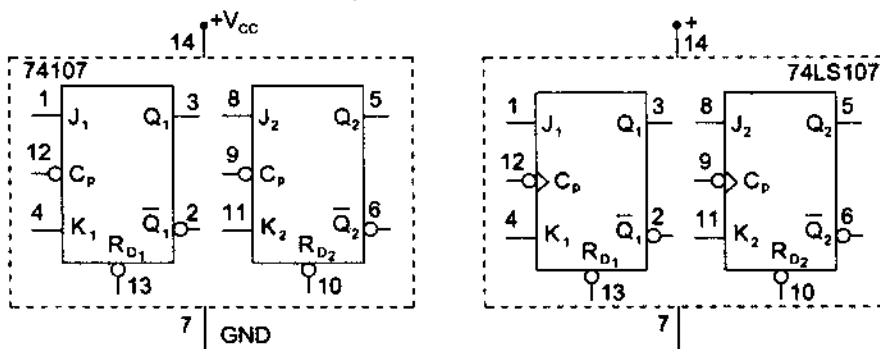
4.8.1. Nhóm IC Flop – Flop

1. IC 7473/74LS 73 bên trong chứa chip 2 x JKFF, tần số làm việc tới 20 MHz/45MHz dòng tiêu thụ tổng cộng 10mA/4mA, nguồn $+5V \pm 5\%$, $T^{\circ}_{\max} = 70^{\circ}C$. Sơ đồ bố trí chân trên hình 4.46 đọc ngược kim đồng hồ từ vị trí khuyết (hình 4.30). 7476/74LS73 kích thích bằng sườn đi xuống.
2. 7474/74LS74A/74S74 có 2 DFF, sơ đồ bố trí chân ở hình 4.47 tần số tối đa 25MHz/33MHz/100MHz. Dòng tiêu thụ 17mA/4mA/30mA. Nguồn $+5V \pm 5\%$, $T^{\circ}_{\max} = 70^{\circ}C$.



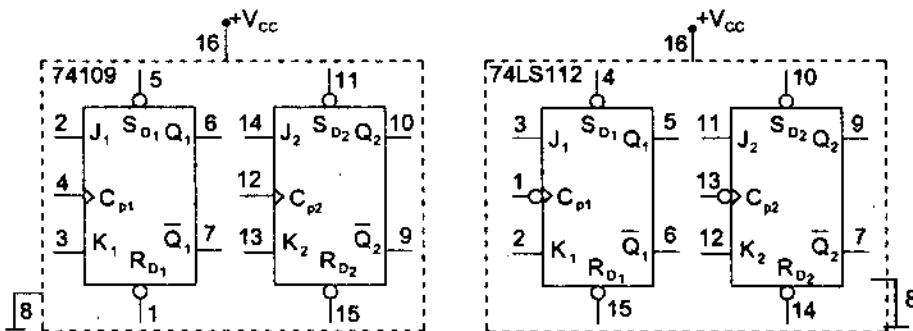
Hình 4.47. Sơ đồ bố trí chân và các cấu trúc khối của IC 7474

3. IC74107/74LS 2xJKFF. Tần số 20 MHz/45MHz dòng tiêu thụ 20mA/4mA
nguồn $+5V \pm 5\%$, $T_{\max}^0 = 70^{\circ}C$. Kích thích ở mức tích cực thấp với loại
74107 và ở sườn xuống với LS107.



Hình 4.48. Sơ đồ bố trí chân và cấu trúc khối IC 74107

4. IC 74109/LS 109A : 33MHz/33MHz ; 2x JK FF 9mA/4mA ; $0 - 70^{\circ}C$;
 $5V \pm 5\%$; kích thích bằng sườn đi lên của xung nhịp.
IC 74LS 112/S112 2JKFF : 45MHz/125MHz ; 4mA/15mA ; $0 - 70^{\circ}C$;
 $5V \pm 5\%$; kích thích bằng sườn đi xuống của xung nhịp.



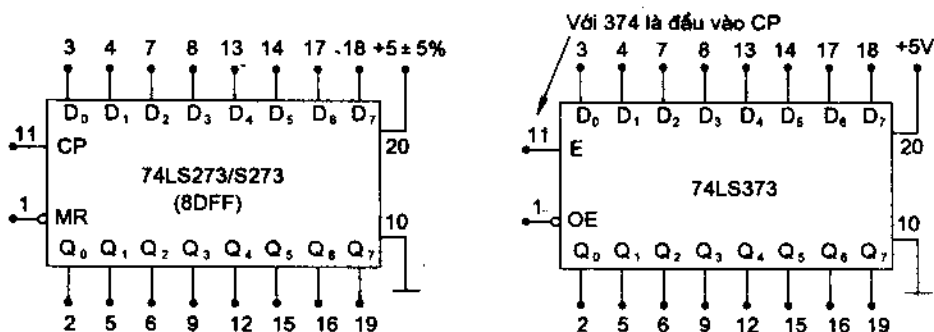
Hình 4.49. Sơ đồ cấu trúc và bố trí chân IC 74109, 74112

5. IC 74LS273/S273 : 40MHz/95MHz ; 17mA/109mA; $5V \pm 5\%$; $0 - 70^{\circ}C$,
kích thích bằng mức tích cực cao của xung nhịp.

IC 74LS364 chứa 8 DFF loại 3 trạng thái ra 50MHz; 42 mA; $5V \pm 5\%$;
 $0 - 70^{\circ}C$ (8 DFF có đầu ra 3 trạng thái).

Chú ý : - Với loại LS364 vị trí MR là đầu vào OE (OUTPUT ENABLE)
để kiểm tra các cửa ra của 8 DFF (3 trạng thái ra).

- Với loại LS 273 là đầu vào MR tích cực mức thấp ($\overline{\text{MR}}$ - Master Reset) có 2 chức năng: đặt trước LOAD hay xoá CLEAR đồng thời cho 8 DFF.



Hình 4.50. Sơ đồ bố trí chân LS 273 và LS373

6. IC 74LS373/ IC74LS374/ S373/ S374

(Thanh chốt/ DFlip Flop loại 8 bit) thời gian trễ 19ns/ 10ns/ 19ns/ 8ns dòng tiêu thụ 24mA/ 105mA/ 27mA/ 116mA.

Đầu vào E cho phép lưu (ghi) đầu vào OE để kiểm tra cổng (loại DFF có đầu ra 3 trạng thái).

4.8.2. IC loại thanh ghi chốt dữ liệu (Latch)

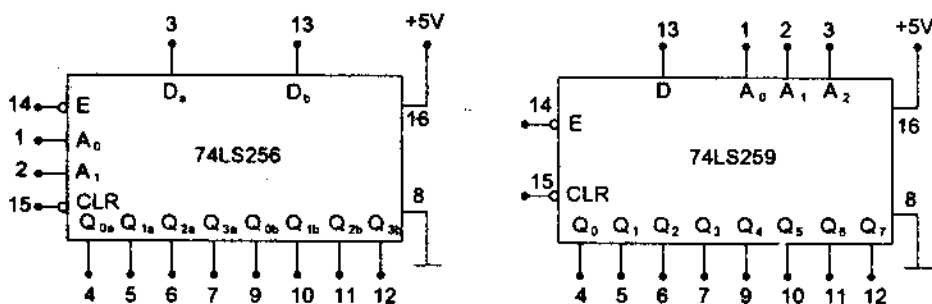
1. IC74LS256 2x4 bit, thời gian trễ 19ns, dòng 22mA nguồn nuôi 5V $\pm$ 5%, nhiệt độ làm việc 0 - 70°C.

2. IC 74LS259 loại 8 bit, trễ 19ns, dòng 22mA, nguồn 5V $\pm$ 5%, 70°C.

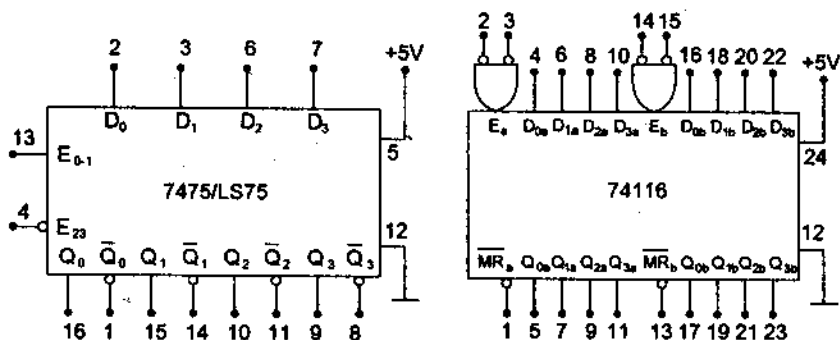
3. IC 7475/ LS75 trễ sườn trước 18ns/ 15ns dòng tiêu thụ 32mA/ 6,3mA

4 x 1 bit, trễ sườn sau 9ns/ 9ns, nguồn +5V $\pm$ 5%, 70°C.

4. IC 74116 2x4bit, trễ 11ns, 50mA, 5V $\pm$ 5%, 70°C.



Hình 4.51. Sơ đồ bố trí chân ICLS 256 và LS259 (bộ chốt dữ liệu)



Hình 4.52. Bố trí chân thanh chốt dữ liệu IC 7475 và 74116

4.8.3. Thanh ghi dịch

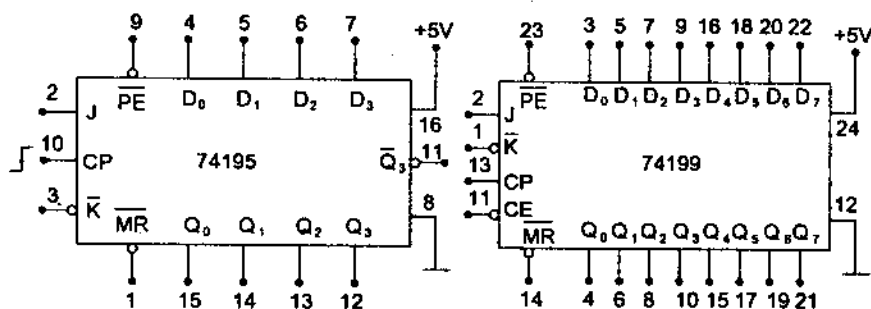
1. IC 74199, $f_{\max} = 35\text{MHz}$, 90 mA, $+5\text{V} \pm 5\%$, $0 - 70^\circ\text{C}$.

Thanh ghi 8 bit song song nối tiếp, dịch phải, nhập dữ liệu song song.

2. IC74195/LS195A/S195 : 39MHz/39MHz/105MHz, 39mA/14mA/70mA, $15\text{V} \pm 5\%$, 70°C .

Thanh ghi 4 bit song song - nối tiếp, dịch phải, nhập dữ liệu song song, có đầu vào J, K, D tăng đầu, Master Reset không đồng bộ (xóa), đầu vào PE (Paralled Enable) khi ở mức 0 ($\overline{\text{PE}} = 0$) sử dụng như : 4DFF đồng bộ; khi ở mức 1 ($\overline{\text{PE}} = 1$) thực hiện chức năng dịch 1 từ Q_0 tới Q_3 (khi $J = 1$) và theo sườn đi lên của C_p .

3. IC74194/LS194A/S194: 36MHz/36MHz/105MHz, 39mA/15mA/85mA, $+5\text{V}$, 70°C . Thanh ghi dịch 4 bit, loại 16 chân IC, dịch phải, dịch trái, chuyển đổi dữ liệu song song - nối tiếp, thời gian trễ từ 20 ns đến 12 ns (với loại S144).



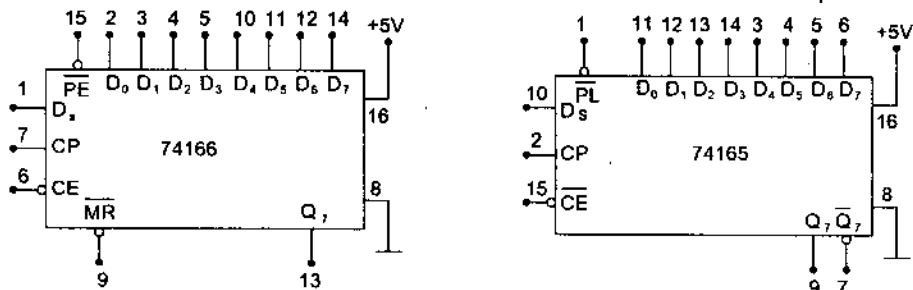
Hình 4.53. Bố trí chân thanh ghi 74195 và 74199

4. IC 74166/ 165 thanh ghi dịch 8 bit vào nối tiếp/ song song, ra nối tiếp.

35MHz ; 90mA; +5V, 70°C; với 165: 26MHz; 42mA.

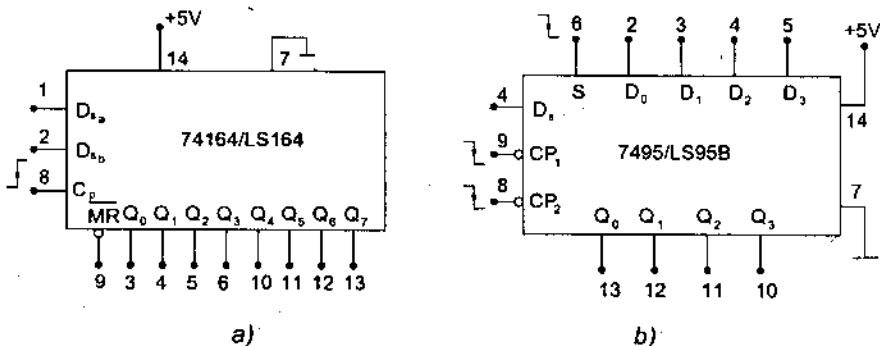
$\overline{CE}$ tích cực mức 0 cho phép CLOCK.

H4.53



Hình 4.54. Bố trí chân của thanh ghi dịch IC 74116, 74165

CP và $\overline{CE}$ phải ở mức thấp trước khi $\overline{PL}$ đầu vào cho phép nhập dữ liệu song song chuyển sang không tích cực.



Hình 4.55. Sơ đồ bố trí chân thanh ghi dịch 74164 a) và 7495 b)

Đặc điểm khi sử dụng IC 74164 (4.55a)

- Dịch phải khi C_p đi lên $\overline{MS}=0$: xóa vào tất cả các đầu CLR. Tích cực mức thấp. D_{s8} , D_{s6} qua cổng AND vào D_0 của DFF₀ (khi nhập 1 dữ liệu D_{s8} thì cho $D_{s6}=1$)
- C_p đi qua 1 bộ đảo vào đồng bộ 8 DFF ở cửa C_p sườn đi xuống.

Đặc điểm khi sử dụng IC 7495 (4.55b)

- Khi $S=1$, sườn đi xuống $\overline{CP2}$ thể hiện được phép nhập $D_0D_1D_2D_3$
- Khi $S=0$, $\overline{CP1}$ được phép nhập dữ liệu nối tiếp D_8 và dịch sau mỗi sườn xuống của $\overline{CP1}$.

Loại 74165 không có $\overline{MR}$, có hai đầu ra nối tiếp Q_7 và $\overline{Q_7}$

$\overline{PL}$: (parallel load) dữ liệu $D_0 - D_7$ nhập khi đồng bộ $\overline{PL}=1$, $D_8=1$ dịch theo nhịp (phải).

5. IC 74164/LS164: Thanh ghi dịch 8 bit vào nối tiếp, ra song song dùng 8DFF, 36MHz/36MHz; 37mA/ 16mA; 5V, 70°C.

IC 7495/LS95B thanh ghi dịch 4 bit vào nối tiếp, ra song song. 36MHz
34mA/13mA.

4.8.4. Các IC đếm thông dụng

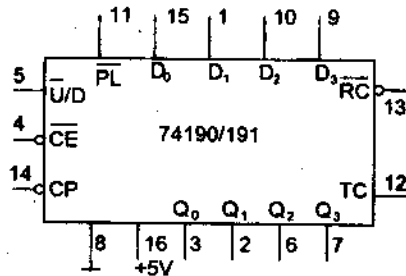
I. IC đếm loại 7490/ LS 90 (30MHz/42MHz, 30mA/ 9mA, +5V $\pm$ 5%, 0 - 70°C).

Chức năng là một bộ đếm thập phân BCD 8421 (IC tương đương : 74LS290, bố trí chân khác). Các đặc điểm chính khi sử dụng :

- $\overline{CP_0}$ tích cực thấp đặt vào Cp của JKFF₀ kích thích bằng sườn đi xuống.
- $\overline{CP_1}$ tích cực thấp đặt vào đồng thời các đầu vào CP của JKFF_{1,2,3} sườn đi xuống.
- MR_1 MR_2 qua 1 cổng NAND tới các đầu vào R_D cả 4 FF.
- MS_1 MS_2 qua 1 cổng NAND tới S_{D0} và S_{D3} (thiết lập trạng thái ra 1001).

và 1111 với 191) khi đó $TC = 1$, $\overline{PL} = 0$ cho phép nhập song song $D_0D_1D_2D_3$ vào bộ đếm (thiết lập trước) lúc đó $Q_i = D_i$.

Bảng trạng thái mô tả cách thiết lập các chế độ làm việc khác nhau của IC đếm theo các đầu vào điều khiển.



a)

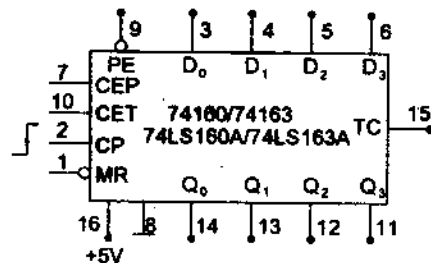
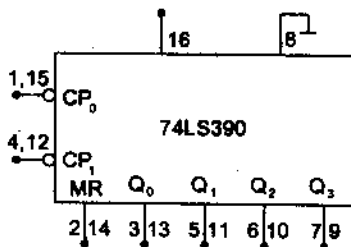
Chế độ làm việc	Đầu vào					Ra	
	$\overline{PL}$	$\overline{U/D}$	$\overline{CE}$	CP	D_n	Q_n	
Nhập song song	0	x	x	x	0	0	
	0	x	x	x	1	1	
Đếm tiến	1	0	0	↑	x	Up	
Đếm lùi	1	1	0	↑	x	Down	
Treo	1	x	1	x	x	—	

b)

Hình 4.57. a) Bố trí chân bộ đếm 10 IC 74190
b) Bảng trạng thái ứng với các chế độ điều khiển

3. IC 74LS390 (55MHz ; 15mA ; +5V, 70°C). Các chức năng IC có thể thực hiện :

Đếm tuần tự thập phân 2 decac, chia 2, 4, 5, 10, 20, 25, 50 hay 100.



Hình 4.58. Sơ đồ chân bộ đếm 74LS390, 74160

4. IC 74160/LS162A đếm thập phân BCD (32MHz, 61mA/19mA ; +5V, 70°C).

IC74161/ LS163A đếm nhị phân 4'bit (32 MHz).

Các đầu vào điều khiển với 160/161: Khi $\overline{MR} = 0$ thao tác xóa : $Q_n = 0$, $TC = 0$

Khi $\overline{MR} = 1$, $\overline{PE} = 0$, $CP = 1$ (sườn lên) $D_n = Q_n$ thao tác nhập dữ liệu song song.

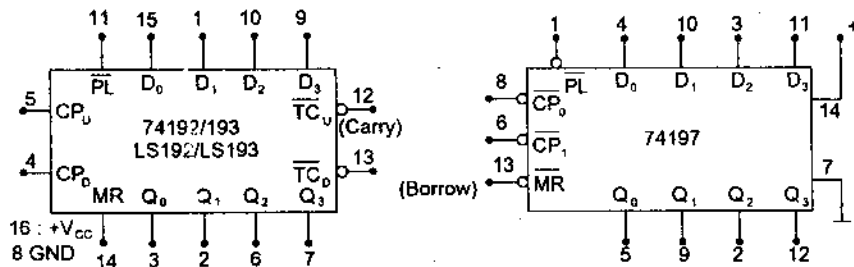
Khi $\overline{MR} = 1$, $\overline{DE} = 1$ thể hiện cấm nhập, đầu vào cho phép đếm $CEP = CET = 1$, $CP = 1$ (sườn lên) thiết lập chế độ đếm.

5. IC 74192/ LS192 bộ đếm BCD. Lập trình được (32MHz ; 65mA/19mA ; 5V, 70°C).

IC 74193/LS193 bộ đếm nhị phân 4 bit (32MHz 65mA/19mA; 5V, 70°C)

Đặc điểm loại IC 74192/ LS192 đếm 10 bước (thập phân BCD)

• Loại IC 74193/ LS193 đếm 16 bước (nhị phân)



Hình 4.59. Bố trí chân bộ đếm 74192, 74197

• Khi $\overline{PL} = 0$: cho phép nhập dữ liệu $D_0D_1D_2D_3$ (đặt trước) vào bộ đếm ($MR = 0$)

• Khi $MR = 1$: xóa $Q_0 = Q_1 = Q_2 = Q_3 = 0$

• Đầu ra TC_D (mượn) = 0 báo tràn dưới (underflow) ; TC_U (nhớ) = 0 báo tràn lên (overflow).

6. IC đếm nhóm CMOS

Chế độ	Đầu vào								Đầu ra					
	MR	$\overline{PL}$	CP_u	CP_0	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3	$\overline{TC}_u$	$\overline{TC}_0$
Xóa	1	x	x	0	x	x	x	x	0	0	0	0	1	0
	1	x	x	1	x	x	x	x	0	0	0	0	1	1
Nhập dữ liệu song song	0	0	x	0	0	0	0	0	0	0	0	0	1	0
	0	0	x	1	0	0	0	0	0	0	0	0	1	1
	0	0	0	x	1	1	1	1	1	1	1	1	0	1
	0	0	1	x	1	1	1	1	1	1	1	1	1	1
Đếm lên	0	1	↑	1	x	x	x	x	Đếm lên				1	1
Đếm xuống	0	1	1	↑	x	x	x	x	Đếm xuống				1	1

$TC_u = CP_u$ (1111)

$TC_0 = CP_0$ (0000)

Hình 4.60. Bảng trạng thái cho 74193

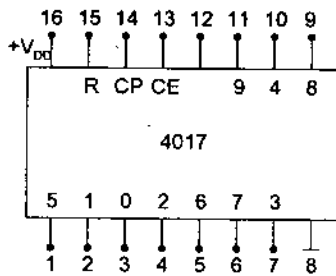
Hình 4.61(abcd) là sơ đồ bố trí chân của các bộ đếm loại CMOS thông dụng.

a) *Bố trí chân bộ đếm 4017*

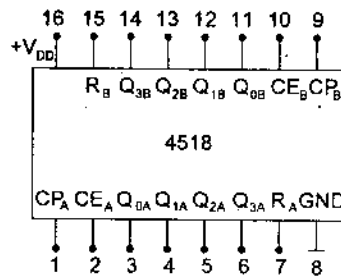
Đặc điểm IC 4017 đếm thập phân, chia 10 theo xung nhịp, 1 trong các đầu ra sẽ chuyển tới mức cao một cách tuần tự (các đầu còn lại đều ở mức thấp) khi $CE(13) = 0$, $R(15) = 0$, sau mỗi xung nhịp CP bộ đếm hoạt động giống phương thức bộ ghi dịch 1 bit sang phải theo từng bước nhịp.

b) *Bố trí chân bộ đếm 4518*

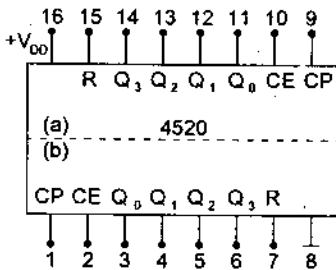
Đặc điểm IC 4518 gồm 2 bộ đếm thập phân, chia 10, chỉ đếm lên, nguồn +3V đến +15V, bố trí chân giống của 4520, nhiệt độ làm việc -55°C đến $+125^{\circ}\text{C}$ loại DFF chủ tố trực tiếp RES (chân 15 với bộ đếm B và 7 với bộ đếm A).



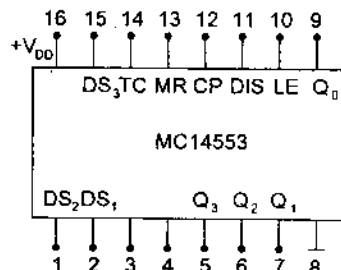
a)



b)



c)



d)

Hình 4.61. Sơ đồ bố trí chân của các bộ đếm loại CMOS

a) 4017 ; b) 4518 ; c) 4520 ; d) 14553

c) *Bố trí chân bộ đếm 4520*

Đặc điểm IC 4520 gồm hai bộ đếm nhị phân, chia 16 sử dụng mã BCD 8421, chỉ đếm lên không đặt trước được, có thể dùng 2 bộ đếm độc lập. Tần số nhịp cực đại 6MHz (+10V nuôi), 2,5MHz (khi nuôi +5V). Nguồn (chân 16) từ +3V đến +15V yêu cầu sườn đi xuống của xung Cp (đếm) và xung cho phép đếm $CE < 10\mu\text{s}$, xung R vào trực tiếp R_D của các FF, xung vào nhịp : $\overline{CP.CE}$ cho mỗi FF.

d) Bố trí chân bộ đếm 14553

Đặc điểm IC MC14553. Bộ đếm 3 digit (3 cấp thập phân). Nguồn +3V đến +18V DS (digit select) chọn digit cho ra kiểu song song (với $Q_0 \dots Q_3$) kết hợp tuần tự (digit BCD 1,2,3) LE =1(latch enable) cho phép chốt, DIS = 1 (disable) cấm đầu vào, MR =1(master reset) xóa toàn bộ FF, TC = 1 cờ báo tràn đếm. Sử dụng thuận lợi trong các nhiệm vụ thiết kế đo tần số, đếm sản phẩm, đếm sự kiện thường gặp trong công nghệ tự động hoá...

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 4

4.1. Tại sao có thể dùng phần tử Flip Flop để lưu giữ các ký số nhị phân ?

Hãy nêu các tính chất quan trọng nhất của một phần tử FF.

4.2. Phân biệt hai loại cấu trúc FF từ các cổng NAND và các cổng NOR về

a) Mạch điện nguyên lý.

b) Tính chất các đầu vào tín hiệu và phản ứng tại đầu ra Q (thông qua các bảng trạng thái).

c) Tại sao phải cấm trạng thái $R = S = 1$ (hay $\bar{R} = \bar{S} = 0$)

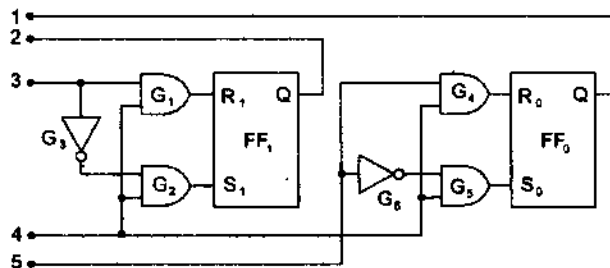
4.3. Tại sao phải có thêm các đầu vào điều khiển ?

Phân biệt điều khiển trực tiếp và điều khiển đồng bộ.

Mạch điện hình 4.5 với thanh ghi 3 bit điều khiển (cho phép hay không cho phép) là loại điều khiển đồng bộ hay điều khiển trực tiếp?

4.4*. Hãy sử dụng IC 74279 hình 4.4 để tổ chức xây dựng một thanh ghi 4 bit $A_3A_2A_1A_0$ vào các RSFF của 74279, kết hợp với 1 đầu vào cho phép E (tham khảo cấu trúc mạch hình 4.5), lưu ý rằng loại RSFF dùng 2 cổng NAND có mức tích cực vào là thấp và nếu NAND có 3 đầu vào thì đầu vào không dùng đến theo hướng dẫn ở chương 2 tiết 2.11 được nối với 1 trong 2 đầu vào còn lại hay nối với $+V_{\infty}$.

4.5. Hãy phân biệt các đầu vào, đầu ra và đầu vào điều khiển trong mạch điện hình B4.5 mô tả một thanh ghi 2 bit dùng RSFF không đồng bộ, loại RSFF trong mạch là cấu trúc hình 4.1a hay hình 4.1b?



Hình B4.5

Hãy thiết lập trạng thái ra khi ở lối vào có tín hiệu cho phép $E = 0$ dữ liệu $A_1 = 1$ và $A_2 = 0$ (giải thích hoạt động trước khi xác định Q).

4.6. Phải cần bao nhiêu cổng NAND để thực hiện tạo ra một RSFF đồng bộ ? Tương tự nếu tạo một DFF cần bao nhiêu cổng NAND ?

- Từ RSFF đồng bộ tạo nên DFF ?
- Từ RSFF không đồng bộ ?

4.7. Thanh ghi có nhiệm vụ gì ?

Nêu 3 cách thực hiện 1 thanh ghi 1 bit nhờ RSFF không đồng bộ, nhờ RSFF đồng bộ và nhờ DFF.

Nói rằng mạch điện thanh ghi không cần dùng mã địa chỉ cho từng ô nhớ của nó có đúng không ?

4.8. Để có 1 thanh ghi/ đọc ghi lại từng 4 bit nhị phân cho 1 từ mã BCD, thể hiện 4 cấp thập phân là 10^3 , 10^2 , 10^1 và 10^0 .

a) Thanh ghi/ đọc này cần bao nhiêu DFF ?

b) Để phân biệt giữa hai thủ tục ghi không đọc và đọc không ghi cần bao nhiêu cổng AND đặt tại các đầu ra của DFF ?

c)* Hãy vẽ cấu trúc thanh ghi này cho 1 từ mã BCD chỉ có 4 bit dữ liệu vào là $A_3A_2A_1A_0$ khi dùng DFF mức tích cực thấp của xung C.

d)* Khi dùng mức tích cực cao của C thì phải thay đổi lại mạch điện đã xây dựng ở câu (c) như thế nào ?

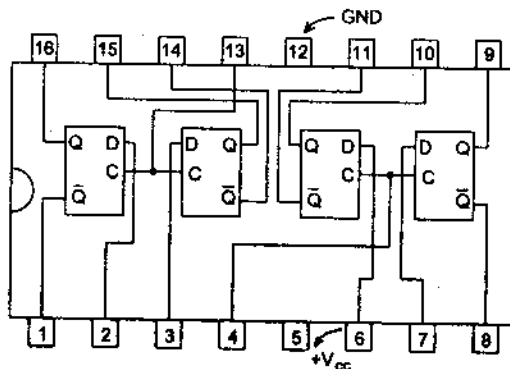
4.9. Hình B4.9 là sơ đồ bố trí chân của IC 7475

a) Có bao nhiêu mạch chốt dữ liệu DFF trong 1 IC ? có bao nhiêu cổng NAND tương ứng?

b) Hãy chỉ các chân đầu vào, đầu ra và đầu vào nhịp của mỗi DFF bên trong IC?

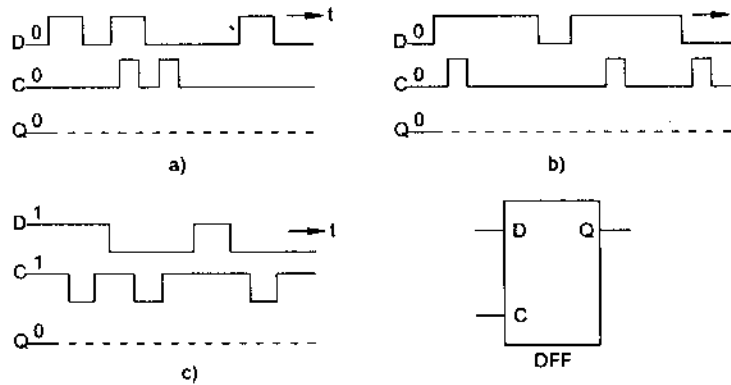
c) Hãy sử dụng IC này giải câu (c)* và (d)* của bài tập 4.8.

d) Nếu dữ liệu chỉ có 3 bit nhị phân có dùng được IC này hay không ?



Hình B49. IC 7475. GND : nối đất 0V ; +V_{cc} nối nguồn dương 5V

4.10. Cho một DFF có các dạng sóng tương ứng ở đầu vào D và đầu ra C như thể hiện trên các đồ thị hình B4.10a, b và c. Hãy xác định giá trị Q dự kiến tại đầu ra trong các trường hợp đã cho, biết rằng lúc $t = 0$, $Q = 0$.



Hình B. 4.10

4.11. Bằng cách nào tạo ra được một Trùng chủ tử (MSFF) từ hai trùng RSFF đồng bộ ? hay từ hai DFF ?

Ưu điểm quan trọng nhất của MSFF là gì ?

Nói rằng giá trị đã nhớ trong MSFF (giá trị cũ) đã mất đi khi người ta nhập một giá trị nhớ mới vào MSFF có đúng không ?

4.12. Phân biệt dữ liệu nối tiếp và dữ liệu song song ?

Có mấy cách chuyển đổi dữ liệu từ nối tiếp sang song song và ngược lại ?

Tại sao việc truyền dữ liệu ở dạng nối tiếp thuận lợi hơn mà truyền dữ liệu song song lại thông dụng hơn nhiều so với nối tiếp ?

4.13*. Hãy phân tích các phương pháp chuyển đổi của thanh ghi dịch đa năng hình 4.25; các đầu vào PR và CL có tác dụng gì ?

a) Kiểu chuyển đổi nối tiếp - nối tiếp: $D_3D_2D_1D_0 = 1010$

b) Kiểu chuyển đổi nối tiếp - song song: $D_3D_2D_1D_0 = 0110$

c) Kiểu chuyển đổi song song - nối tiếp: $B_3B_2B_1B_0 = 1101$

Yêu cầu nhấn mạnh tác dụng điều khiển trực tiếp của các đầu vào PR và CL.

4.14. Bộ đếm vòng là gì ? Làm thế nào để nó tự xóa ?

Hãy thực hiện 1 bộ đếm vòng gồm 1 thanh ghi dịch có 3 ô nhớ loại DFF có mức tích cực thấp của xung nhịp.

4.15. a) Đặc điểm của 2 phương pháp kích thích mạch FF ?

Ưu nhược điểm của các phương pháp này là gì ?

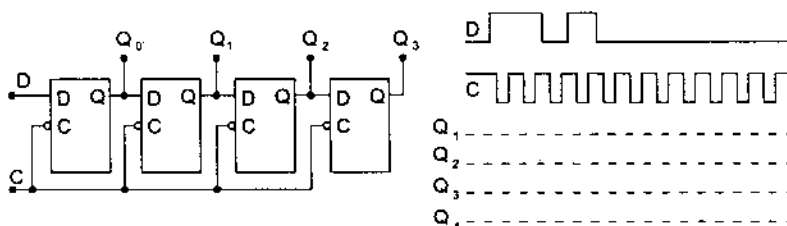
b) Hãy vẽ 4 trường hợp ký hiệu quy ước cho các phần tử số kích thích bằng mức tích cực (cao, thấp) và kích bằng sườn xung (lên, xuống).

c)* Cho đồ thị thời gian minh họa 4 phương pháp nêu ở câu (b).

4.16. a) Hãy vẽ sơ đồ một mạch chốt dữ liệu chủ tố sử dụng 11 cổng NAND, C tích cực ở mức thấp, có đầu ra $\bar{Q}$.

b) Câu hỏi tương tự xây dựng MSFF dùng 9 NAND (kiểu RSFF).

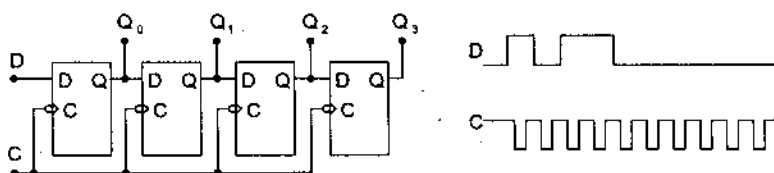
4.17. Hãy vẽ dạng sóng đầu ra Q_i cho thanh ghi dịch 4 bit hình B4.17 có dữ liệu D và nhịp C cho trước.



Hình B4.17

4.18. Câu hỏi tương tự cho mạch điện thanh ghi dịch hình B4.18.

Vẽ Q_0 ? Q_1 ? Q_2 ? Q_3 ?



Hình B4.18

Nếu chuyển đổi FF thành các phần tử phản ứng bằng sườn đi lên thì đồ thị các Q_0 Q_1 Q_2 Q_3 sẽ thay đổi như thế nào?

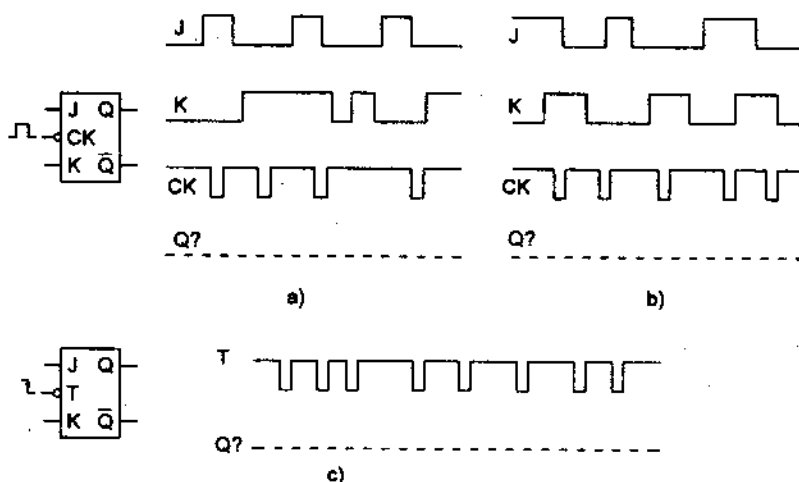
4.19. Tại sao gọi JKFF là Trơ bập bênh ? trạng thái nào đặc trưng cho phương thức hoạt động này ?

Hãy chứng minh tính chất đa năng của JKFF bằng cách từ nó tạo ra các trơ khác: DFF, RSFF không đồng bộ, RSFF đồng bộ và TFF.

4.20. Hãy vẽ một thanh ghi dịch nối tiếp 3 bit dùng JKFF loại IC 74LS73 (xem hình 4.46) và giải thích hoạt động của thanh ghi khi có một dữ liệu 3 bit $D = D_2D_1D_0 = 101$ nối tiếp xuất hiện ở đầu vào đồng bộ với xung nhịp CK. Nếu chu kỳ nhịp CK là T_0 thì sau bao lâu tại đầu ra song song và đầu ra nối tiếp sẽ nhận lại đầy đủ các dữ liệu $D_2D_1D_0$ đã ghi vào.

4.21. a) Hãy xác định dạng của tín hiệu ra Q trên đồ thị hình B4.21a, b khi tại đầu vào của JKFF tác động các tín hiệu J, K và CK đã cho.

b) Câu hỏi tương tự cho hình B4.21c với tín hiệu vào là T tác động tới TFF.



Hình B4.21. (chú ý JKFF ở hình (a) và (b) kích bằng mức tích cực thấp, T ở hình c kích bằng sườn xuống)

4.22. Có 1 dữ liệu $D_0 \dots D_7 = 1001\ 1101$ muốn ghi vào thanh ghi dùng IC 74LS273 (xem hình 4.50).

a) Có mấy cách để ghi dữ liệu này vào thanh ghi ?

Hãy nêu trình tự cần thiết để thực hiện việc ghi song song dữ liệu đã cho vào thanh ghi này.

b) Khi kiểm tra chân IC thấy : Các chân theo thứ tự 19, 2, 5, 6, 9, 16, 12, 15 các trạng thái 10101101 phép ghi ở câu (b) đã đúng dữ liệu cần ghi chưa ? Có sai ở đâu không ? (Hãy sửa lại chỗ sai nếu có).

4.23. Muốn ghi dữ liệu 96 (trong hệ thập phân) vào thanh ghi dùng IC 74116 (xem hình 4.52)

a) Hãy nêu thủ tục thực hiện thao tác ghi.

b) Viết trạng thái ra của các đầu ra của thanh ghi khi đã ghi đúng kết quả.

c) Nếu dùng IC 7475 để thực hiện việc ghi dữ liệu đã cho có được không, nếu được hãy ghi kết quả các trạng thái ra dự kiến sẽ có nếu phép ghi thực hiện đúng ?

4.24. Thực hiện việc ghi dữ liệu $D_0 \dots D_7 = 0110\ 1001$ vào thanh ghi dịch dùng IC74165 (xem hình 4.54).

a) Có mấy cách ghi dữ liệu này vào ?

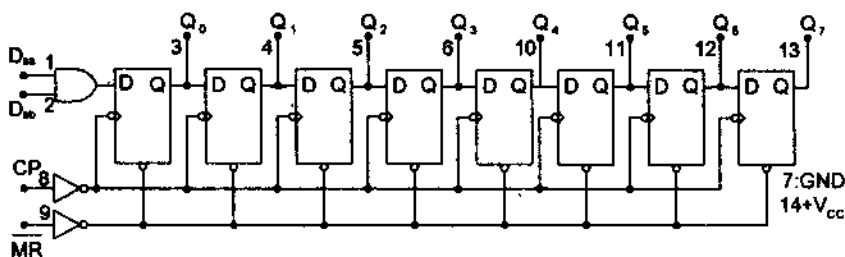
b) Có mấy cách lấy dữ liệu (đọc) ra khỏi thanh ghi ?

c) Sau khi đã ghi đúng, lúc cần đọc dữ liệu thì đầu vào $\overline{CE}$ ở mức nào ? Sau 5 nhịp của CP, đầu ra ở mức 0 hay 1 ? là giá trị của dữ liệu D_i nào ?

4.25. a) Có thể thực hiện ghi song song dữ liệu cho ở bài tập 24 vào IC 74164 (xem hình 4.55) được không ? Có thể dùng đồng thời 2 đầu vào D_{sa} và D_{sb} để đưa dữ liệu đã cho vào thanh ghi này không ?

b) Nếu ghi được thì sau bao nhiêu xung CP thì dữ liệu đã cho được ghi xong vào thanh ghi ?

c) Có thể dùng đầu ra Q_7 để lấy toàn bộ dữ liệu đã ghi ra được không ? Có cách nào khác để lấy các dữ liệu ra ? (xem hình B4.25).



a)

Chế độ	Đầu vào				Đầu ra	
	$\overline{MR}$	CP	D_{sa}	D_{sb}	Q_0	$Q_1 - Q_7$
Xoá (Reset)	0	x	x	x	0	0 - 0
Dịch	1	↑	l	l	0	$q_0 - q_6$
	1	↑	l	h	0	$q_0 - q_6$
	1	↑	h	l	0	$q_0 - q_6$
	1	↑	h	h	1	$q_0 - q_6$

b)

Chú ý:

l : ở mức thấp khi xung Cp áp vào đang từ thấp lên cao

h : ở mức cao khi xung Cp áp tới đang từ thấp lên cao

Hình B4.25. a) Mạch điện thanh ghi dịch IC 74164/ LS1648 bit vào nối tiếp ra song song

b) Bảng chức năng chế độ làm việc

4.26. a) Hãy phân biệt bộ đếm nối tiếp (kích thích không đồng bộ) và bộ đếm song song (kích thích đồng bộ) ?

b) Khi sử dụng DFF hay JKFF để xây dựng 1 bộ đếm, nếu có 8 trigơ sẽ đếm được tối đa bao nhiêu xung đếm Cp (trong hệ đếm nhị phân) ? Nếu để đếm trong hệ BCD 8421 thì số xung đếm tối đa là bao nhiêu ?

c) Người ta cần chia 1 dãy có chu kỳ T_0 thành một dãy có chu kỳ $64 T_0$ thì cần bao nhiêu FF ?

d) Nếu cần chu kỳ ra là $50 T_0$ thì làm thế nào ?

4.27*. a) Có thể dùng 9 Trigơ JKFF để thực hiện một bộ đếm hệ 24 được không ?

b) Nếu được hãy viết bảng trạng thái của bộ đếm này, dựa trên cách sử dụng 3 bộ đếm hệ tám ghép liên tiếp nhau.

c) Vẽ sơ đồ mạch điện của một bộ đếm 8 kích thích đồng bộ dùng JKFF.

d) Vẽ sơ đồ khối của mạch điện bộ đếm (bộ chia) 24 dựa trên kết quả của câu (c) đã có.

4.28. Sử dụng IC đếm 4518 để đếm số thập phân hàng trăm có được không ? nếu được thì bằng cách nào ? hãy vẽ sơ đồ khối thực hiện phép đếm trên (xem hình 4.61b).

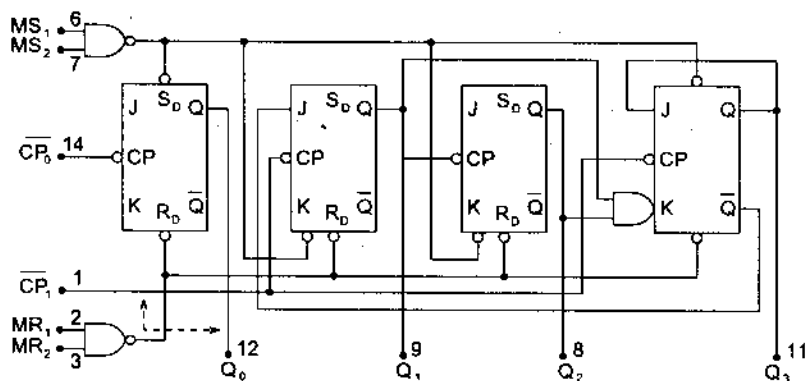
4.29. a) Cần bao nhiêu IC đếm 7490 để đếm số thập phân hàng nghìn ?

b) Khi sử dụng 7490 để đếm trong hệ 10 cần chú ý đến đặc điểm gì để nối thêm dây ở ngoài mới thực hiện được ?

c) Nếu dùng 7490 thực hiện một bộ đếm 5 có được không ? làm như thế nào nếu được ?

d) Tương tự nếu dùng IC 7490 chỉ để thực hiện 1 bộ đếm hệ 2 thì làm như thế nào ? đếm được mấy bit nhị phân ?

e) Bằng cách nào có được một bộ đếm hệ 20 khi dùng IC 7490 ? (xem hình B4.29).



Hình B.4.29a. Mạch điện IC đếm thập phân 7490/ 74LS90 (mã BCD 8421)

Vào RESET/SET				Ra			
MR ₁	MR ₂	MS ₁	MS ₂	Q ₀	Q ₁	Q ₂	Q ₃
1	1	0	x	0	0	0	0
1	1	x	0	0	0	0	0
x	x	1	1	1	0	0	1
0	x	0	x	đếm			
x	0	x	0	đếm			
0	x	x	0	đếm			
1	0	0	x	đếm			

b)

Hình B4.29

b) Bảng chức năng hoạt động của 7490 ;

c) Bảng chân lý khi thực hiện đếm hệ 10

Chú ý đầu ra Q₀ phải nối tới CP₁

Xung vào đếm	Đầu ra			
	Q ₀	Q ₁	Q ₂	Q ₃
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1

c)

4.30*. Nếu sử dụng 4 IC4520 có thể thực hiện được phép chia tối đa là bao nhiêu ? Có bao nhiêu phép chia 1 dãy xung nhịp có thể thực hiện được khi dùng 4IC loại này?

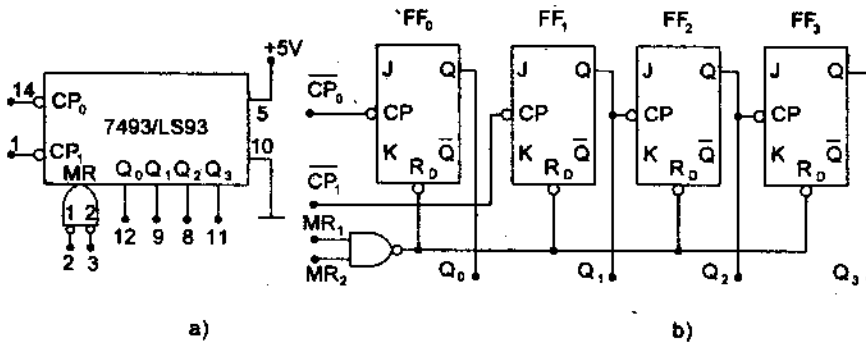
4.31. Cần chia 1 dãy xung nhịp có tần số là 1kHz = 10³Hz thành dãy xung có tần số là 1Hz, đã có IC 74LS90 (xem hình 4.58) hay IC 4518 (xem 4.61b).

a) Cần dùng bao nhiêu IC loại tương ứng ?

b) Nêu cách ghép nối chúng để thực hiện ?

c) Làm thế nào để chia tiếp dãy xung kết quả cho 60 để nhận được 1 xung có tần số là 1/60Hz hay chu kỳ là 60s ?

4.32. Hãy vẽ mạch điện thực hiện các bộ đếm hệ 4, hệ 8, hệ 10 và hệ 12 dùng IC đếm 7493 có sơ đồ chân và mạch điện cấu trúc cho trên hình B4.32 dưới đây (bộ đếm nhị phân 4 bit).



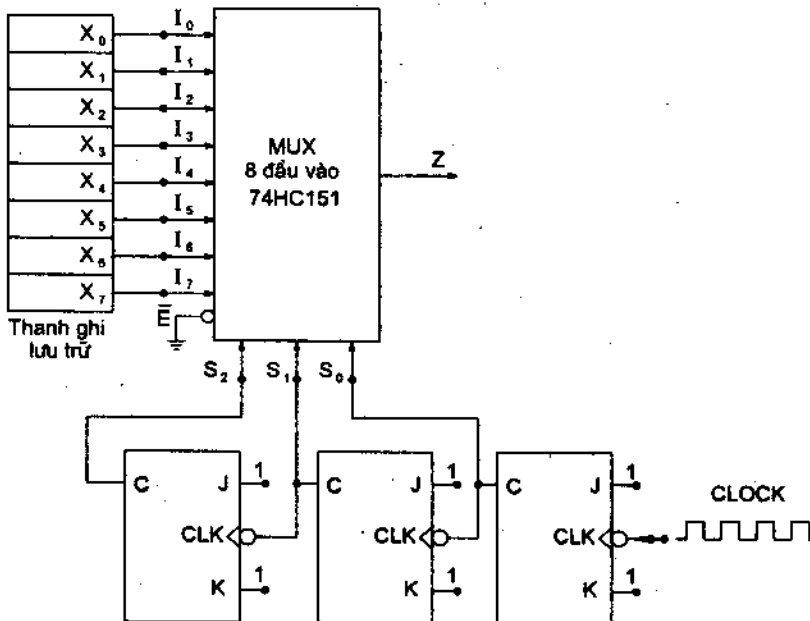
a) b)

Hình B.4.32. a) Bố trí chân của IC 7493
b) Mạch điện cấu trúc của IC 7493 Khi $MR_1 = MR_2$

Khi $MR_1 = MR_2$: xoá $Q_0 = Q_1 = Q_2 = Q_3 = 0$

Khi $MR_1, MR_2 = 0$: chế độ đếm theo sườn đi xuống của CP₀ hay CP₁ (tích cực mức thấp)

4.33. Cho mạch điện hình B4.33. Hãy nhận dạng và phân tích nhiệm vụ của mạch qua quan hệ hàm Z ở lối ra với các đầu vào dữ liệu từ X₀ đến X₇ và đầu vào địa chỉ S₂ S₁ S₀.



Hình B4.33

Chương 5

CÁC BỘ NHỚ BÁN DẪN

5.1. KHÁI NIỆM CHUNG VỀ BỘ NHỚ

5.1.1. Khái niệm

Bộ nhớ là một thiết bị (cấu kiện) có khả năng lưu trữ thông tin (nhị phân). Muốn sử dụng bộ nhớ, trước tiên ta phải ghi dữ liệu và các thông tin cần thiết vào nó, sau đó lúc cần thiết phải gọi (lấy) được dữ liệu hoặc thông tin đã ghi trước đây ra sử dụng. Thủ tục ghi vào và đọc ra phải được kiểm soát chặt chẽ, tránh nhầm lẫn nhờ định vị chính xác từng vị trí nhớ và nội dung của nó theo một mã địa chỉ duy nhất để tìm kiếm để người sử dụng biết rằng: vị trí nhớ xác định đó còn đang trống (chưa được sử dụng tới) trong thủ tục ghi vào và biết rằng dữ liệu thông tin cần tìm đang nằm ở đâu? vị trí nào ? Khi làm thủ tục lấy ra (đọc).

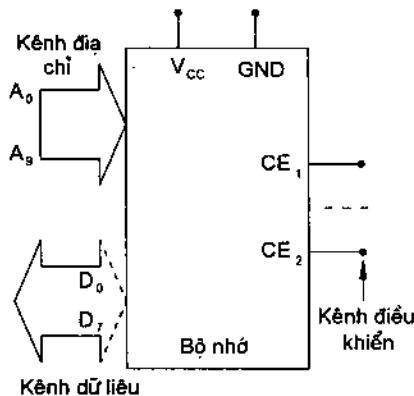
5.1.2. Phân loại

Theo chức năng đã nêu trên, có 2 loại bộ nhớ cơ bản là bộ nhớ cho phép cả ghi vào và đọc ra (Read - Write - Memory gọi tắt là RAM – Random Access Memory – theo thuật ngữ bộ nhớ truy nhập ngẫu nhiên) và bộ nhớ chỉ cho phép đọc (Read – Only - Memory gọi tắt là ROM). Có thể phân loại bộ nhớ theo một số tính chất và đặc điểm của nó, trong đó quan trọng hơn cả là dung lượng và tốc độ (xuất nhập) : Dung lượng của bộ nhớ đánh giá khả năng (hay sức chứa) thông tin của nó. Dung lượng nhớ được hiểu là tổng số bit nhị phân có thể lưu giữ bên trong bộ nhớ. Có thể ước tính dung lượng một bộ nhớ qua tổng số các đầu vào, đầu ra của nó thực hiện việc trao đổi thông tin giữa bộ nhớ với bên ngoài. Hình 5.1 biểu thị tổng quát cấu trúc khối của một bộ nhớ.

Bộ nhớ giao tiếp với bên ngoài thông qua 3 kênh (Bus) thông tin cơ bản: kênh địa chỉ, kênh dữ liệu và kênh điều khiển. Thông qua kênh điều khiển, kênh dữ liệu và kênh địa chỉ phải được phối hợp làm việc một cách nhịp

nhau (đồng bộ) với nhau để kiểm soát việc xuất nhập dữ liệu một cách chính xác, đúng thủ tục (hợp lệ) với mục đích điều khiển hay hỗ trợ các thiết bị khác trong một hệ thống kỹ thuật số.

– Với ví dụ hình 5.1 bộ nhớ có 8 đầu vào dữ liệu từ D_0 đến D_7 thể hiện kích cỡ 1 từ mã nhị phân được xuất nhập là 8 bit (ra vào đồng thời kiểu song song) sẽ được ghi hay đọc vào 1 vị trí nhớ của nó gồm 8 ô nhớ (mỗi ô lưu một bit).



Hình 5.1. Cấu trúc tổng quát của 1 bộ nhớ. Ngoài 2 chân V_{cc} và GND bộ nhớ tiếp xúc với thiết bị ngoài nhờ 3 kênh thông tin (chú ý đường đứt nét dành cho RAM)

– Với bộ nhớ hình 5.1 có 10 đầu vào địa chỉ ký hiệu từ A_0 đến A_9 sẽ cho biết có thể lưu 1 từ 8 bit vào một trong số 2^{10} địa chỉ vị trí nhớ hợp lệ chứa bên trong nó, từ mã địa chỉ $(0000000000)_2 = (000)_{16}$ đến mã địa chỉ $(1111111111)_2 = (3FF)_{16} = (1024)_{10}$. Khi đó ta quy ước gọi bộ nhớ hình 5.1 có dung lượng nhớ là 1K ($1K = 10^3 \approx 1024$) (việc gọi này chỉ là quy ước cho dễ nhớ mà không làm ảnh hưởng tới nội dung số lượng các vị trí nhớ bên trong nó vì đây là cách gọi gần đúng hợp lý nhất) 1K đọc là 1 kilo các từ 8 bit nghĩa là bộ nhớ đang xét có dung lượng 1Kbyte. Nếu kênh địa chỉ có 16 bit từ A_0 đến A_{15} thì số lượng vị trí nhớ sẽ là $2^{16} = 65.536$, mỗi vị trí nhớ được 1 từ 8 bit và ta có dung lượng của bộ nhớ này là 64K(byte).

5.1.3. Làm tăng dung lượng của một bộ nhớ

Việc tăng dung lượng bộ nhớ là một biện pháp quan trọng để mở rộng năng lực và phạm vi làm việc, các chức năng hoạt động của một hệ thống số. Tuy nhiên thường gặp một mâu thuẫn là việc tăng dung lượng bằng cách tạo ra các chip IC nhớ có sức chứa lớn (làm tăng số lượng địa chỉ các vị trí nhớ) sẽ kèm theo việc tìm kiếm chúng sẽ vất vả và lâu hơn và do vậy sẽ làm chậm tốc độ của việc nhập/xuất thông tin chứa trong bộ nhớ và kéo theo làm chậm tốc độ làm việc của các thiết bị khác. Một biện pháp có hiệu quả hơn là phân cấp quản lý thông tin cho các bộ nhớ:

- Các thông tin (dữ liệu hoặc lệnh) cần thiết ngay cho công tác của các thiết bị số liên quan sẽ được lưu giữ trong một bộ nhớ được gọi là bộ nhớ chính. Bộ nhớ chính có đặc điểm dung lượng không cần lớn lắm (đến vài chục kilobít) nhưng tốc độ tìm kiếm địa chỉ các vị trí nhớ và xuất (nhập) nội dung nhanh (10^{-8} s) và thường được bố trí sát bộ phận xử lý dữ liệu hay nhận thông tin từ bộ nhớ tới.

- Với các nội dung thông tin chưa cần sử dụng ngay (ví dụ kết quả của một công việc vừa xử lý xong cần chờ các kết quả tiếp theo để xử lý tiếp tục...) sẽ được lưu giữ vào các bộ nhớ phụ, chúng mang tính chất như một thư viện chứa các nội dung thông tin tạm trú, chờ đến lúc cần thiết thì sử dụng. Các bộ nhớ phụ cần có dung lượng lớn (tới hàng ngàn tỷ bit) nhưng tốc độ tìm kiếm địa chỉ và xuất/nhập nội dung thông tin chậm hơn nhiều (tới $ms = 10^{-3}s$).

- Biện pháp thường gặp trong việc tăng dung lượng là đặt song song các chip IC nhớ cỡ bit (xuất nhập đồng thời các bit) hay đặt song song các chip IC nhớ cỡ từ (8 bit, 16 bit...) và việc xuất nhập được xử lý với từng chip IC khi được lệnh cho phép nó tích cực (xem hình 5.5 mô tả cách ghép các ROM song song).

5.1.4. Thời gian thâm nhập bộ nhớ (access time)

Một thông số quan trọng quyết định tốc độ làm việc của bộ nhớ từ lúc có lệnh đến lúc thực hiện xong một thao tác đối với một nội dung dữ liệu hay thông tin chuyển qua bộ nhớ, đó là thời gian thâm nhập bộ nhớ. Thời gian thâm nhập bộ nhớ gồm 2 phần:

- Thời gian cần thiết để tìm kiếm định vị 1 vị trí nhớ của nó (nếu còn trống thì sẽ ghi thông tin vào, còn nếu là đang lưu giữ thông tin thì sẽ đọc nội dung ra).

- Thời gian cần thiết để đưa thông tin nhập vào vị trí này (nếu ghi) hay lấy nội dung thông tin đang giữ ở vị trí này xuất ra khỏi bộ nhớ (nếu đọc).

Có thể phân loại bộ nhớ theo một vài tham số khác như : cách tổ chức ma trận ô nhớ theo vị trí, phương pháp vật lý để nhớ dữ liệu 1 bit nhị phân, cấu trúc các bộ giải mã địa chỉ bên trong, cách thức giao tiếp vật lý của bộ nhớ với các kênh thông tin tiếp xúc với nó... một tham số cũng được đánh giá là chi phí giá thành của IC nhớ tính theo tiêu chuẩn: mức chi phí/1bit nhớ.

5.2. BỘ NHỚ CHỈ ĐỌC ROM (READ - ONLY - MEMONY)

5.2.1. Khái niệm về ROM

Để một máy tính điện tử tiến hành các hoạt động bình thường của mình dù là một hoạt động đơn giản nhất, cần phải cung cấp cho máy trước tiên là các lệnh từng bước tiến hành. Tập hợp danh sách các lệnh này được gọi là chương trình. Việc nhập một chương trình nào đó từ đĩa hay bàn phím vào máy tính thậm chí cũng phải tuân theo sự điều khiển của một chương trình đặc biệt gọi là chương trình khởi động (hay chương trình mồi - bootstrap programm) thường trú bên trong máy, được cài sẵn ở một bộ nhớ chỉ đọc (ROM). Bộ nhớ ROM có các thuộc tính sau:

- Chỉ cho phép đọc nội dung đã được ghi sẵn từ trước chứa trong nó. Nội dung này do người điều hành thiết kế lập trình sẵn và ghi vào nó bằng một phương pháp đặc biệt.

- Người sử dụng về nguyên tắc không thể hoặc khó lòng thay đổi nội dung thông tin đã ghi nhớ trong ROM.

- Nội dung được ghi trong ROM có tính chất cố định, không bị mất đi theo thời gian hay do sự cố mất nguồn năng lượng cung cấp chung cho toàn hệ thống số trong đó có ROM. Tính chất nội dung cố định (thường trú) này phân biệt rất rõ với các bộ nhớ có nội dung nhớ mang tính chất tạm thời (tạm trú) và sẽ bị mất đi trong bộ nhớ khi đã lấy ra hay khi mất điện ở nguồn chính mà không có biện pháp bảo vệ dữ liệu này bằng các nguồn thay thế.

Do những thuộc tính trên, ROM được dùng khi:

- Chứa các chương trình vận hành máy tính, chương trình hỗ trợ việc lập trình hay các chương trình giám sát việc điều hành theo các bước nghiêm ngặt cố định.

- Các chương trình cài đặt cho máy tính chuyên dụng chỉ thực hiện một nhóm các công việc có giới hạn, được lập đi lập lại có quy luật do công nghệ tự động hoá đòi hỏi (ví dụ như tự động điều khiển động cơ cho một ôtô...)

- Lưu giữ các bảng biểu quan trọng được thường xuyên sử dụng tới như là công cụ cho việc tính toán xử lý của máy tính và các thiết bị liên quan.

5.2.2. Cấu trúc chung của ROM (hình 5.2)

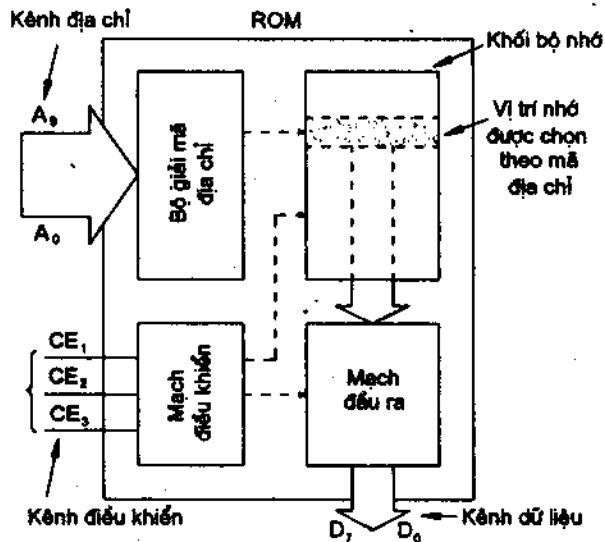
ROM bao gồm có 4 khối cơ bản : mảng bộ nhớ chứa các vị trí nhớ và đơn vị cơ bản là ô nhớ trong 1 vị trí nhớ, mạng mạch điều khiển tiếp nhận các tín hiệu vào từ kênh điều khiển để nhận biết phải làm việc gì ? Bộ giải mã địa chỉ để xác định vị trí nhớ trong mảng bộ nhớ (theo mã địa chỉ nhận được, tìm ra vị trí nào được cho phép tiếp xúc với bên ngoài) và mạch đầu ra

để đưa nội dung đã tìm thấy trong vị trí nhớ tới kênh dữ liệu gửi tới các thiết bị có liên quan cần tiếp nhận nội dung này.

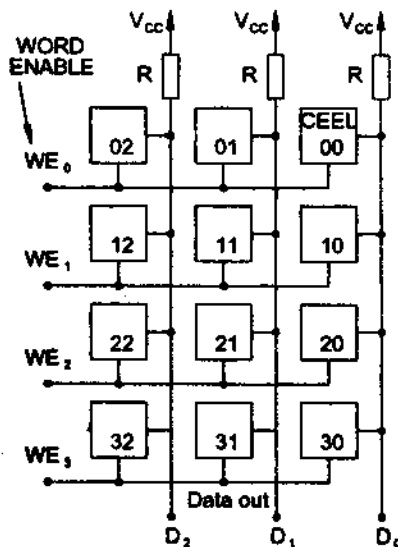
5.2.3. Khối bộ nhớ có phần tử cơ bản là 1 ô nhớ nhị phân

Ô nhớ nhị phân (ví dụ dùng 1 DFF) có chức năng nhớ (lưu) 1 trong 2 trạng thái 0,1 là giá trị nhị phân, tập hợp một nhóm các ô nhớ xếp theo hàng tạo thành 1 vị trí nhớ có khả năng lưu giữ 1 từ 8 bit (hay 16... bit) nhị phân. Ví dụ, 1 bộ nhớ có dung lượng $1K \times 8$ chứa tổng cộng $8.1024 = 8192$ ô nhớ và có khả năng nhớ 1024 byte. Kiểu sắp xếp trên đây gọi là cấu trúc bộ nhớ kiểu mảng (array) tuyến tính. Hình 5.3 biểu diễn một cấu trúc bộ nhớ kiểu mảng tuyến tính 4×3 (dung lượng chứa được 4 từ mã mỗi từ 3 bit). Trình tự đọc (xuất) 1 từ đã được nhớ như sau:

Để xuất (đọc) 1 từ cụ thể, trước tiên đầu vào điều khiển cho phép từ WE (WORD ENABLE) được kích hoạt (tích cực ở mức cao hay mức thấp tùy theo bản chất vật lý cấu tạo của ô nhớ). Ô nhớ phải có đầu ra loại hở cực colectơ hay loại đầu ra 3 trạng thái như đã xét ở chương 2, tất cả



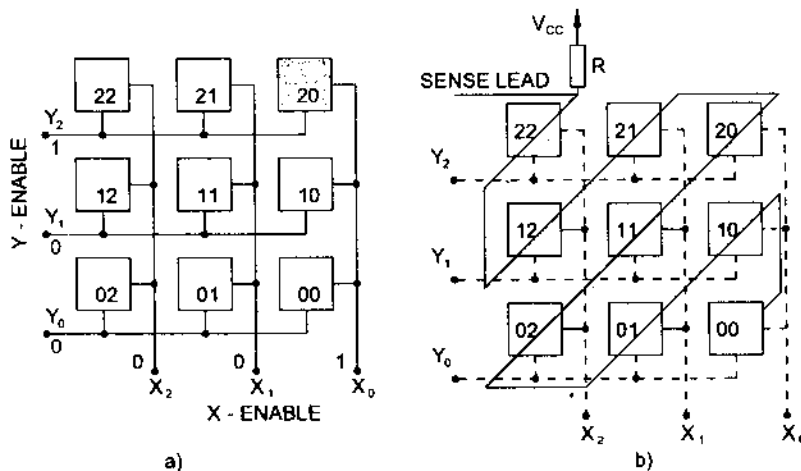
Hình 5.2. Cấu trúc 4 khối cơ bản bên trong của 1 bộ nhớ chỉ đọc ROM.



Hình 5.3. Cấu trúc bộ nhớ kiểu máy tuyến tính 4 từ (hàng) x 3bit (cột)
(Mạch ra của 1 ô nhớ là loại colectơ để hở nên có điện trở R kéo lên)

các ô nhớ trong một cột xác định có đầu ra nối chung tới một đầu ra dữ liệu, vì 1 từ (hình 5.3) chỉ có 3 bit nên có 3 đầu ra dữ liệu $D_2 D_1 D_0$ tương ứng với 3 cột, 4 hàng tương ứng với 4 từ được điều khiển theo các xung $WE_0 WE_1 WE_2$ và WE_3 tức là mỗi từ phải có đầu vào WE riêng, đây là một hạn chế của cấu trúc mảng tuyến tính vì khi số lượng vị trí địa chỉ tăng lên với các bộ nhớ dung lượng lớn cấu trúc mạch giải mã địa chỉ sẽ rất phức tạp.

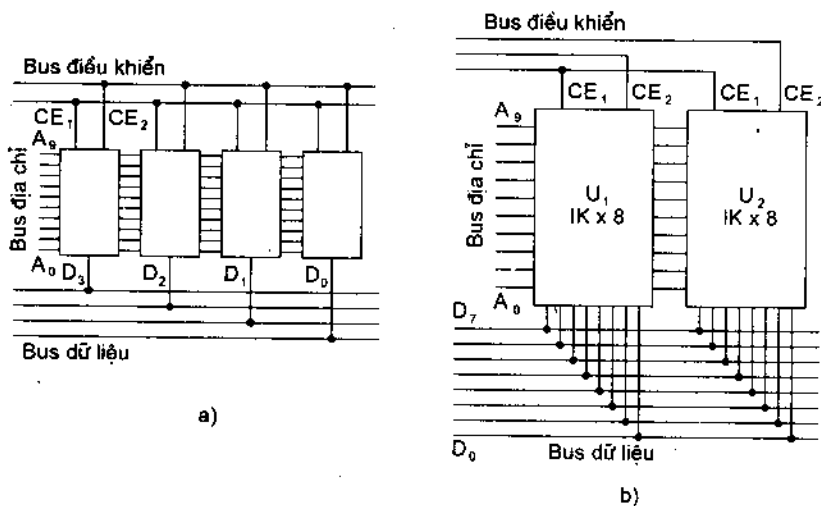
Dạng cấu trúc thứ hai của bộ nhớ là tổ chức kiểu ma trận, thích hợp với các bộ nhớ có dung lượng lớn được cho trên hình 5.4. Hình 5.4a cho thấy kết cấu mảng ma trận (3 x 3) các ô nhớ. Ví dụ này đủ tổng quát cho một ma trận n hàng ký hiệu từ Y_0 đến Y_{n-1} , m cột ký hiệu từ X_0 tới X_{m-1} và chứa tất cả tổng cộng $(n \times m)$ ô nhớ nhị phân trong đó, bộ nhớ có ma trận $(n \times m)$ ô nhớ có dung lượng là $(n \times m)$ bit nhớ. Mỗi ô nhớ của ma trận (gọi là 1 nút của ma trận) có hai đầu vào cho phép và khi chúng tích cực đồng thời thì ô mới xuất được một bit đã lưu trong nó. Ví dụ, để kích hoạt ô nhớ ký hiệu 20 các đầu X_0 và Y_2 phải ở tích cực, chẳng hạn ở mức cao ($X_0=Y_2=1$), giá trị trên đầu cho phép phải là 2 tổ hợp: $X_2X_1X_0=001$ và $Y_2Y_1Y_0=100$. Với hai tổ hợp cho phép hàng Y_i và cột X_j này chỉ một ô nhớ duy nhất có địa chỉ (ij) được chuyển lên tích cực. Hình 5.4b mô tả đường vẽ cắt mặt của tất cả các ô nhớ gọi là đường SENSE (thụ cảm) nhấn mạnh tới việc chỉ 1 dây trên được sử dụng để dò tới ô tích cực theo hướng bit và dữ liệu chứa trong 1 ô nhớ sẽ được đọc ra chỉ nằm dọc theo đường này, mỗi ô nhớ phải là một phân tử có 3 trạng thái ra ở cực góp: mức 0, mức 1 và Hi.Z (xem chương 2 phần từ 3 trạng thái). Dung lượng của bộ nhớ hình 5.4b là 9×1 (mỗi từ chỉ dài 1 bit,



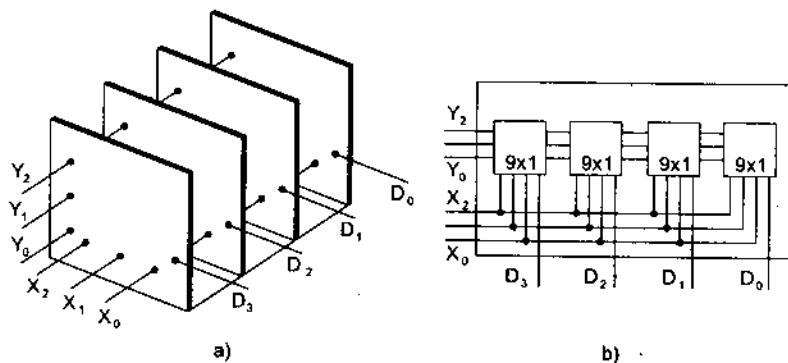
Hình 5.4. Cấu trúc bộ nhớ kiểu mảng ma trận hướng bit, tín hiệu điều khiển tìm đến từng điểm nút ma trận là 1 ô nhớ đơn lẻ.

có 9 từ được nhớ vào 9 ô nhớ) chúng được tổ chức theo bit. Muốn tạo ra 1 từ 4 bit ta sẽ nối 4 mảng ma trận như vậy song song với nhau như trên hình 5.5a thể hiện, ở đây mỗi mảng ma trận chứa 1024 ô nhớ (1Kb), như vậy với cách ghép song song này dung lượng bộ nhớ tổng cộng sẽ là (1Kx4) bit. Còn ở hình 5.5b trước tiên tổ chức ma trận 1K x 8bit bằng cách đấu song song 8 chip ROM loại 1Kx1 được U_1 có dung lượng (1Kx8) bit, sau đó ghép song song hai mảng này sẽ được 1 ma trận có dung lượng nhớ là (2Kx8) bit.

Trong ví dụ hình 5.4b nối song song 4 mảng 9 x 1 ta được bộ nhớ có dung lượng là 9 x 4 bit được biểu diễn trên hình 5.6a trong không gian 2 chiều và quy về mặt phẳng có mạch hình 5.6b.



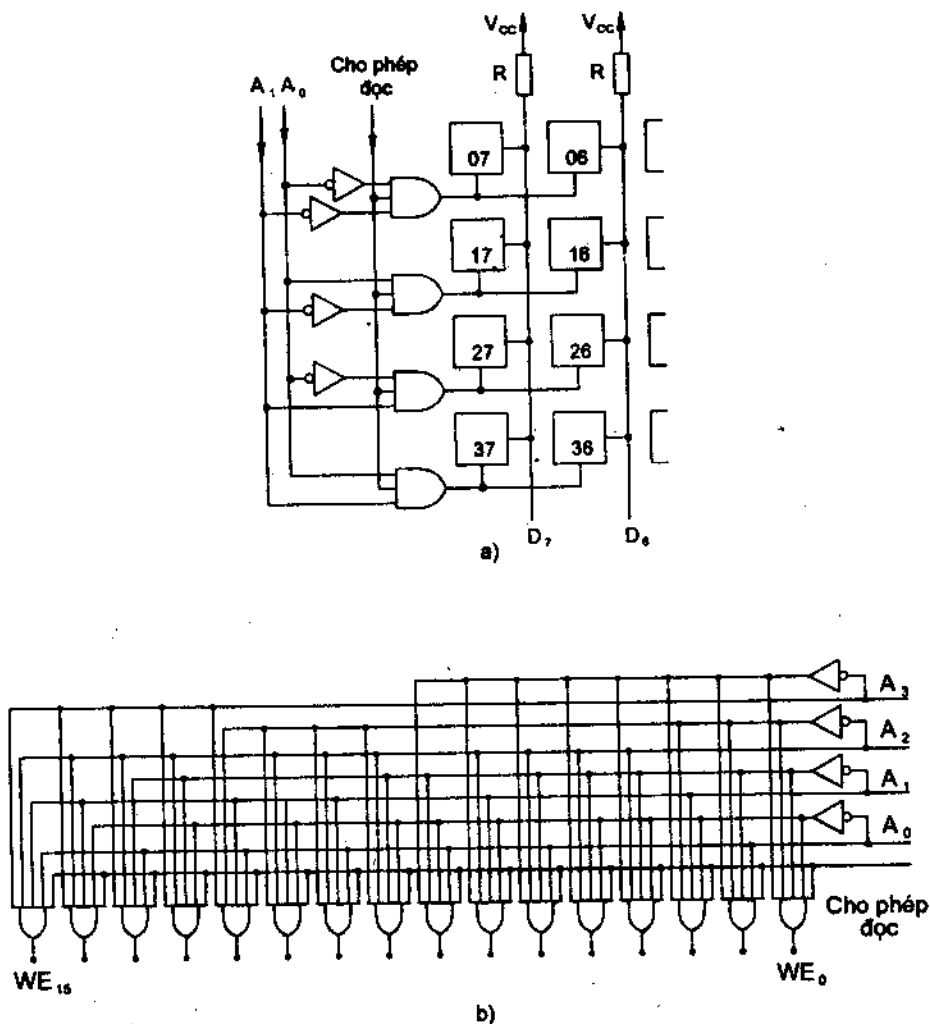
Hình 5.5. a) 4 chip Rom 1K x 1 nối song song để tạo thành bộ nhớ 1K x 4
b) Cách đấu song song kép (1K x 8) x 2 để tăng dung lượng bộ nhớ lên 2K x 8



Hình 5.6. Ghép song song 4 mảng 9 x 1 với nhau để tạo thành 1 mảng ô nhớ với dung lượng 9 x 4 bit

5.2.4. Bộ giải mã địa chỉ

Bộ giải mã địa chỉ là giao diện giữa kênh địa chỉ và khối bộ nhớ có khả năng truyền rất nhiều địa chỉ trên một số ít đường truyền. Địa chỉ nhị phân áp vào phải được giải mã trước khi tác động tới mảng ô nhớ.

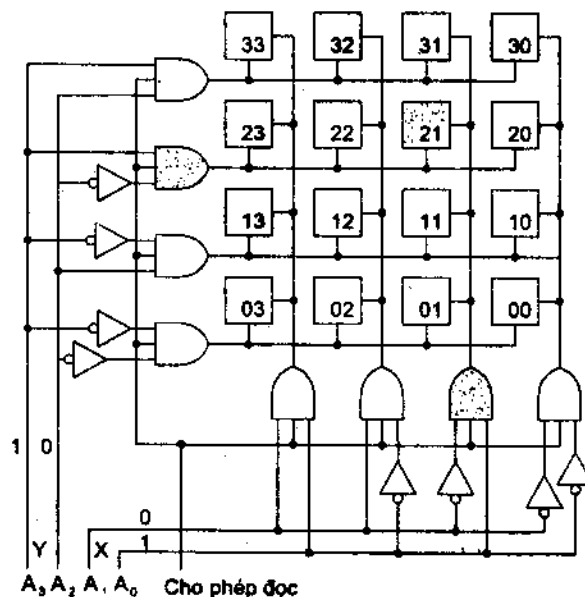


Hình 5.7. Các bộ giải mã địa chỉ

- a) Bộ giải mã địa chỉ 1 trong 4 với 2 bit địa chỉ A_1, A_0 cho 32 ô nhớ (4 hàng x 8 cột)
- b) Bộ giải mã địa chỉ 1 trong 16 với 4 bit địa chỉ A_3, A_2, A_1, A_0 cho thấy khi số bit địa chỉ tăng lên (bộ nhớ dung lượng lớn ví dụ 1K) mạch sẽ phức tạp tới mức nào.

* Trường hợp với cấu trúc bộ nhớ dạng mảng tuyến tính, xét bộ giải mã hình 5.7a A_1 và A_0 là các đầu vào địa chỉ 4 đầu cho phép WE_0 WE_1 WE_2 và WE_3 là các đầu ra của bộ giải mã, ứng với 1 tổ hợp mã A_1A_0 chỉ có 1 đầu ra WE lên tích cực. Hình 5.7b là ví dụ bộ giải mã có 4 bit địa chỉ $A_3A_2A_1A_0$ với việc chọn 1 trong 16 tổ hợp của các bit địa chỉ lấy ra 1 đầu WE cho phép (từ chỉ số WE_0 đến WE_{15}).

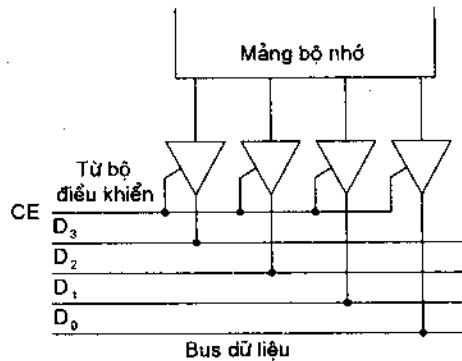
* Với trường hợp bộ nhớ có cấu trúc ô nhớ dạng ma trận, việc tổ chức mạch giải mã địa chỉ theo hàng và cột của ma trận trở nên đơn giản hơn nhiều. Với việc chọn 2 bit địa chỉ A_1A_0 cho địa chỉ cột của ma trận và 2 bit còn lại A_3A_2 cho địa chỉ hàng của ma trận, như vậy chỉ cần 8 cổng AND và 8 cổng NOT đã tạo ra được một địa chỉ hàng i cột j xác định ứng với 1 ô nhớ trong ma trận được chọn ra. Ví dụ $A_3A_2A_1A_0 = 10\ 01$, khi đó vị trí hàng 10 cột 01 (hàng 2 cột 1 trong biểu diễn thập phân của i, j), tức là ô thứ 21 được chọn (chuyển lên tích cực).



Hình 5.8. Bộ giải mã địa chỉ cho mảng ma trận ô nhớ (4 x 4) bit (không vẽ mạch ra của ma trận ô nhớ theo đường-cắm thụ)

5.2.5. Mạch đầu ra của bộ nhớ

Mạch có nhiệm vụ kết nối dữ liệu đã chọn xuất hiện tại khối bộ nhớ với kênh dữ liệu vào lúc thích hợp. Như mạch ra hình 5.9 thể hiện, bộ đếm 3 trạng thái được sử dụng với mục đích ngắt nối kết giữa bộ nhớ với kênh dữ liệu khi bộ nhớ không tích cực (không được chọn làm việc), các phân tử đếm 3 trạng thái phải có hệ số tải cao để kích thích kênh dữ liệu với vô số tải khác nối tới kênh.



Hình 5.9

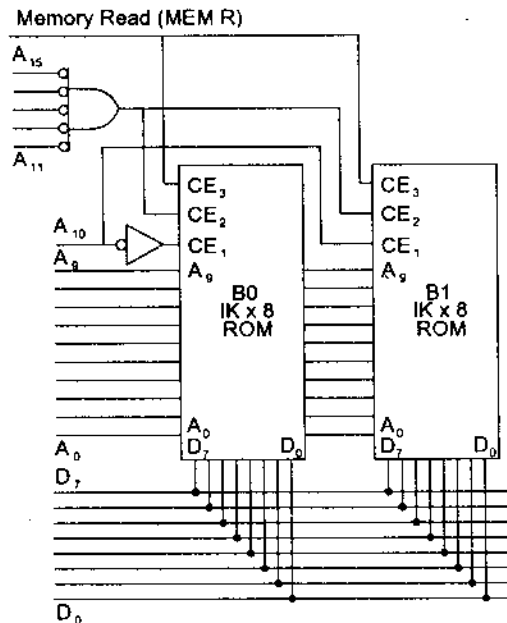
Phần tử đầu ra 3 trạng thái dùng làm mạch đầu ra của ROM. Tín hiệu CE từ bộ điều khiển cho phép (hay không) kết nối giữa bộ nhớ và BUS dữ liệu

5.2.6. Mạch điều khiển trong ROM có chức năng khá đơn giản, ví dụ với cấu trúc khối hình 5.2 chỉ có 3 đầu vào điều khiển CE_1 , CE_2 và CE_3 (viết tắt của Chip Enable) hay còn gọi là CS (Chip Select - chọn chip) khi cả 3 tín hiệu này đều tích cực thì ROM mới được phép hoạt động (cần 1 cổng AND hay NAND 3 đầu vào). Có thể bao gồm nhiều chức năng được giao cho CE đảm nhiệm :

- * Chọn chip cần đọc (khi bộ nhớ có nhiều chip ghép song song nhau, như ví dụ hình 5.5).

- * Định giờ cho phép việc nối kết ROM với kênh dữ liệu.

- * Tham gia vào tiến trình giải mã địa chỉ để mở rộng khả năng kiểm soát và chọn địa chỉ cho chính xác.



Hình 5.10. Dùng đầu vào điều khiển CE để chọn chip của ROM

Hình 5.10 minh họa 3 chức năng điều khiển mà tín hiệu CE tham gia :

Bộ nhớ có chứa 2 chip IC nhớ loại 1K (gọi là có 2 trang nhớ – page) hay hai khối nhớ. Bus địa chỉ của máy tính có 16 bit từ A_0 đến A_{15} và chúng sẽ xác định 2K ô nhớ, các bit địa chỉ A_0 đến A_9 đã được nối song song và được giải mã đồng thời ở bên trong 2 chip này. Tuy vậy chỉ có chip nào có đồng thời 3 đầu vào CE tích cực áp tới mới được phép xuất dữ liệu ra, (ví dụ trong hình 5.10 bit A_{10} sẽ quyết định chip nào trong 2 chip 1K sẽ ở tích cực). Điều này có nghĩa là chip B_0 sẽ có khối địa chỉ nhị phân là 11 bit, từ 0 00000 00000 đến 0 11111 11111, còn khối chip B_1 sẽ có các mã địa chỉ ô nhớ từ 100000 00000 đến 111111 11111. Hệ thống địa chỉ này có được cho 2 chip B_0 và B_1 với điều kiện các bit địa chỉ cao hơn A_{10} (A_{11} A_{12} A_{13} A_{14} A_{15}) đều ở 0 để đảm bảo không nhầm lẫn, cho thêm cổng NOR và như vậy CE_2 chỉ tích cực khi $A_{11} = \dots = A_{15} = 0$. Tất cả các địa chỉ khác có bit địa chỉ từ $A_{11} = 1$ trở lên đến $A_{15} = 1$ đều cho CE_2 không tích cực, nghĩa là B_0 và B_1 là các chip không được chọn tham gia vào quá trình xử lý.

Với 16 bit địa chỉ có tối đa $2^6 \cdot 2^{10} = 64K$ hay 64 chip 1K được gán địa chỉ, các bit A_{10} đến A_{15} được tổ hợp để chọn 1 trong 64 chip 1K này cho lên tích cực để xuất dữ liệu, còn A_9 đến A_0 chọn vị trí địa chỉ một ô nhớ cụ thể trong chip đã được chọn lên tích cực này. Đầu điều khiển CE_3 dùng để định giờ hoạt động cho ROM, khi tín hiệu MEMR xuất hiện (mức cao chẳng hạn) dữ liệu đã chọn được quyền xuất ra Bus dữ liệu.

5.3. CÁC DẠNG ROM THƯỜNG GẶP

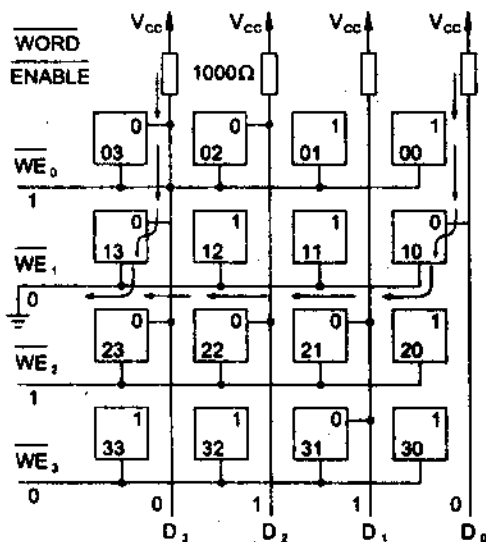
Điểm khác biệt chủ yếu của ROM là cấu tạo vật lý chế tạo các ô nhớ trong mảng ô nhớ của nó. Phần này đề cập đến một vài phương pháp lập trình sẵn cho ROM, tuy nhiên cũng chỉ quan tâm hơn tới các loại đã được cải tiến và sử dụng nhiều hiện nay.

5.3.1. ROM lập trình bằng mặt nạ (Mask - programmed - ROM)

ROM lập trình mặt nạ là một dạng cơ bản được nạp dữ liệu nhớ ngay trong quá trình chế tạo theo ý định của người đặt hàng sau khi đã cung cấp danh sách dữ liệu cần lưu trong ROM cho người sản xuất, thường giai đoạn lập trình ở lúc đặt tập hợp mạch dẫn cuối cùng cho ROM. Cấu trúc của các ô nhớ trong ROM mặt nạ thể hiện trên hình 5.11, đường đậm nét từ ô nhớ nối ra dây đọc là các ô nhớ được nối dây thể hiện đã nhớ bit 0, còn ô không

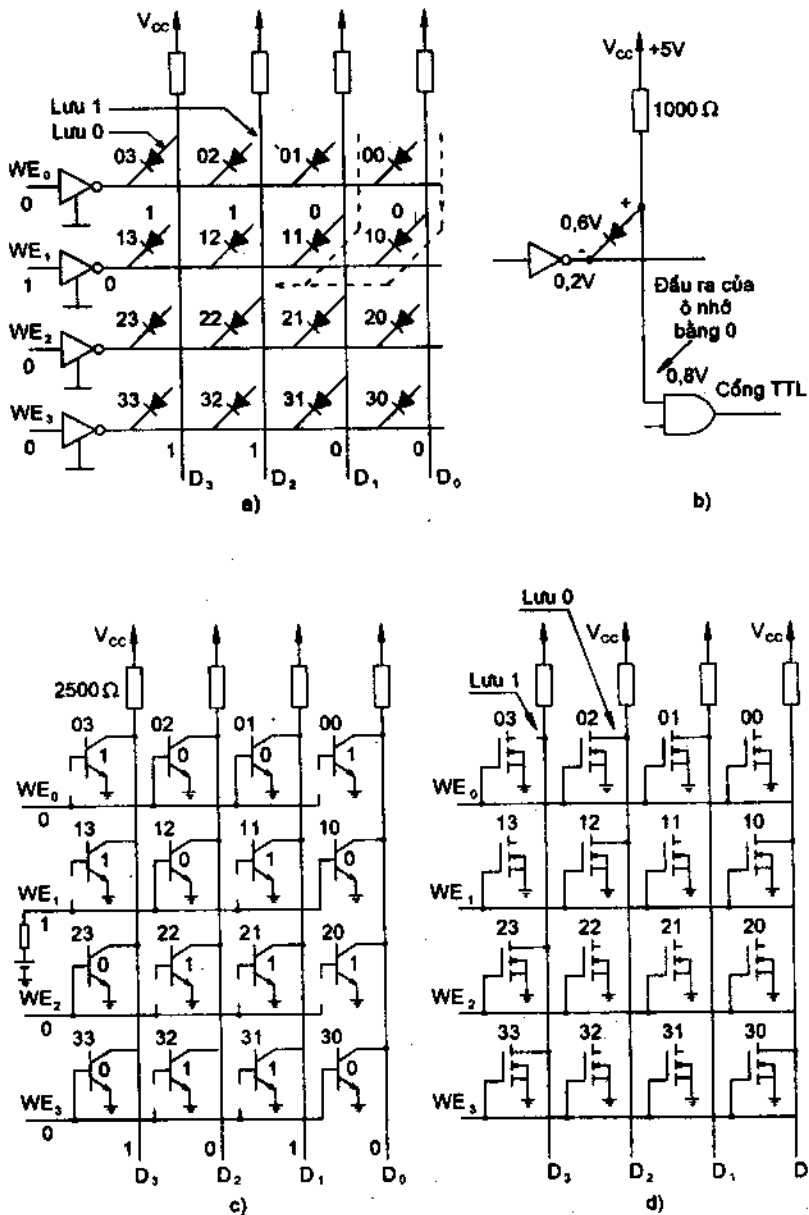
được nối dây thể hiện đã lưu bit 1 (là các lớp SiO₂ cách điện theo công nghệ gắn giống công nghệ Plana chế tạo BJT, FET và IC)

Để đọc số lưu trong một ô nhớ xác định (ví dụ hàng thứ 2) cho tín hiệu $\overline{WE}_1 = 0$, các hàng khác không được đọc vì $\overline{WE}_0 = \overline{WE}_2 = \overline{WE}_3 = 1$. Nếu xem mỗi ô là “một ngăn mạch” thì với dòng qua các ô 13 và 10 làm mức điện thế đầu ra $D_3 = D_0 = 0$ (đọc mức 0 ghi ở ô 13 và 10) trong khi



Hình 5.11. Mô hình ROM mặt nạ

đó các ô 12 và 11 tuy vẫn được $\overline{WE}_1 = 0$ cho phép nhưng khi lập trình không có dây nối tới nhánh cột (đã ghi 1) nên mức thế $D_2 = D_1 = 1$ (mức cao vì không có dòng qua $R = 1k\Omega$). Tuy nhiên phần tử ngăn mạch của một ô nhớ như mô tả yêu cầu phải không dẫn dòng theo chiều ngược lại, nếu không mức cao $D_2 = 1$ sẽ bị kéo xuống 0 do dòng qua ô 02 và ô 03, vì thế người ta thay phần tử dây ngăn mạch lần lượt bằng các điôt, các tranzitor lưỡng cực và tranzitor đơn cực (FET) có tác dụng van đối với dòng điện ở chế độ nối mạch (chỉ cho dòng qua theo 1 hướng) như thể hiện ở các hình 5.12a, c và d. Hình 5.12 b giải thích nhược điểm của cách nối bằng điôt làm mức 0 có điện thế cỡ 0,8V không chấp nhận với cổng TTL phía sau. Hình 5.12c là loại ROM mặt nạ dùng tranzitor lưỡng cực (BJT), các BJT có bazơ để hở là ứng với việc lưu bit 1 còn bazơ được nối là ứng với việc lưu bit 0 tại ô nhớ. Tại hình 5.12a khi có tín hiệu $\overline{WE}_1 = 1$ các điôt D_{11} và D_{10} dẫn điện cho phép đọc ra mức $D_1 = D_0 = 0$ ($\overline{WE}_1 = 0$ nối tới katôt của chúng) còn các điôt 13 và 12 do hở mạch nên $D_3 = D_2 = 1$ (mức thế cao) lệnh $\overline{WE}_1 = 1$ cho phép đọc hàng 2 được từ mã 1101 đã ghi trong ROM. Trên hình 5.12c ứng với $\overline{WE}_1 = 1$ dữ liệu đã nhớ trong ROM được đọc ra là 1010. Các ROM có dung lượng lớn thường được chế tạo theo dạng mạch hình 5.12d do FET đòi hỏi năng lượng ít hơn, chiếm ít chỗ hơn nên trên 1 chip tạo được nhiều ô lớn hơn, hệ số tải đầu ra lớn, đầu vào nhỏ cũng là ưu điểm nổi bật của các cổng dùng MOS, tuy nhiên ảnh hưởng của nhiễu tĩnh điện lớn nên dễ làm mất hay sai thông tin đã được ghi, cần có các biện pháp công nghệ chặt chẽ đảm bảo an toàn thông tin được lưu trữ.



Hình 5.12

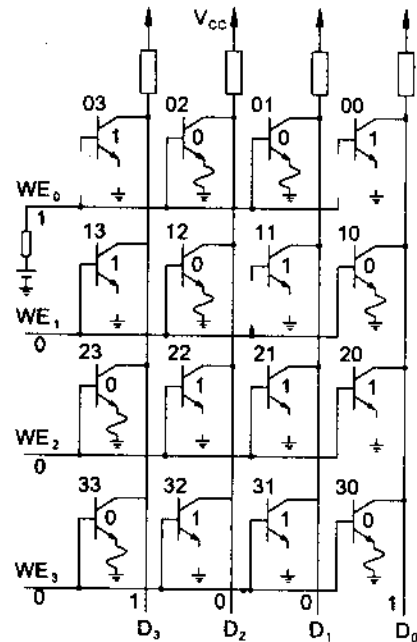
- a) ROM mật nă dùng diốt nối mạch các bit nhớ 0;
b) Nhược điểm của dùng diốt không phối ghép được với TTL mức thấp;
c) ROM mật nă dùng BJT, khi cực B được nối là ô lưu giữ bit 0;
d) ROM mật nă dùng FET nối mạch khi cực mắg được nối mạch ô tương ứng đang lưu bit 0.
Dòng đầu các ô nhớ 03 02 01 00 lưu thông tin 1001, tương tự dòng tiếp lưu thông tin 1010, dòng thứ 3 đã nhớ từ 0111 và dòng cuối cùng có dữ liệu 0110 trong các ô nhớ 33 32 31 30 tương ứng.

5.3.2. ROM cho phép lập trình (PROM- programmable ROM)

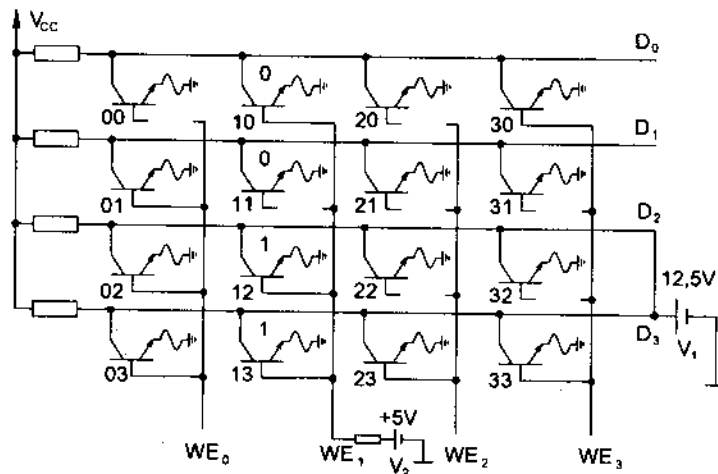
Để khắc phục các nhược điểm của ROM mặt nạ là tính năng động kém do thời gian chế tạo phụ thuộc vào thời gian đặt hàng, hầu như không thay đổi được nội dung, giá thành tương đối cao. PROM do người sử dụng nó tự lập trình. Hình 5.13a là cấu tạo một PROM nối chì cực phát của tất cả các tranzitor lưỡng cực trong các ô nhớ khi chế tạo xong. Người sử dụng muốn nạp dữ liệu nhớ cố định, ví dụ là 4 từ 4 bit :

$$D_{03} D_{02} D_{01} D_{00} = 1001$$

$$D_{13} D_{12} D_{11} D_{10} = 1010$$



Hình 5.13 a) PROM nối cấu chì ở E



Hình 5.13b) Tự lập trình cho PROM đối với từ ở vị trí các ô 13, 12, 11, 10 (hàng có chỉ số 1 hay còn gọi là từ (1) gồm 4 bit 1100)

$$D_{23} D_{22} D_{21} D_{20} = 0111$$

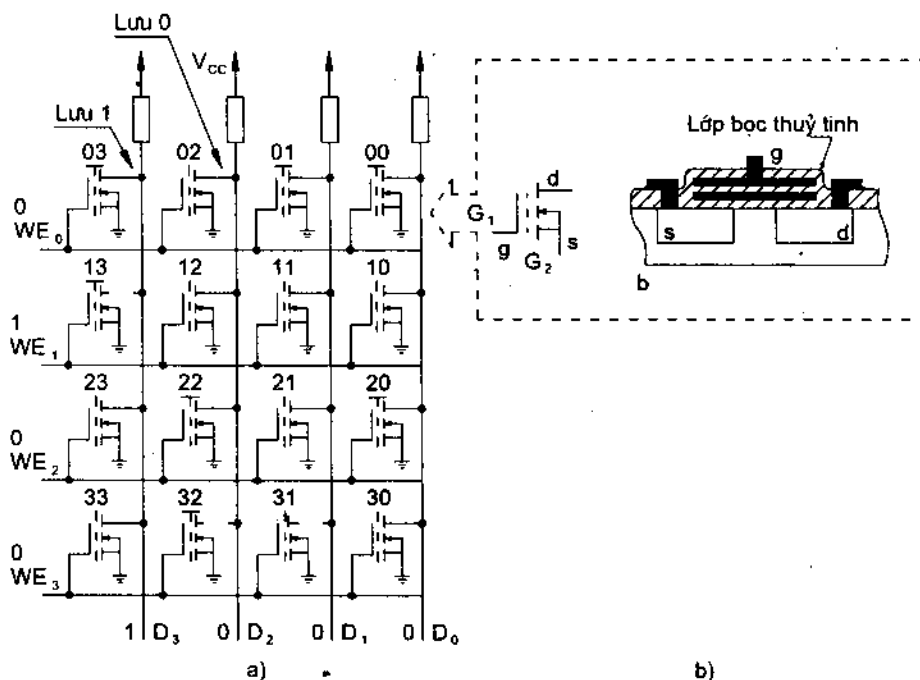
$$D_{33} D_{32} D_{31} D_{30} = 0110$$

Khi đó tất cả các bit giá trị 1 được nạp vào các ô nhớ bằng cách làm đứt cầu chì ở cực E của tranzitor tương ứng với ô đó. Khi nuôi +5V dòng cực E không đủ làm đứt chì, cần dùng áp lớn hơn mức chuẩn ở những chỗ có chọn lọc như đã nêu. Hình 5.13b mô tả cách lập trình cho PROM để đưa dữ liệu cần nhớ 1100 vào từ 13, 12, 11, 10 với nguồn phụ là $+V_1 = 12,5V$ và $V_2 = +5V$ đưa tới WE_1 tạo mức cho phép ghi của từ (1), PROM chỉ cho phép lập trình 1 lần không làm lại được nên chỉ một sai sót nhỏ là hỏng cả chip.

5.3.3. EPROM (chip ROM cho phép xóa và lập trình lại erasable PROM)

1. Cách lập trình (nạp) cho EPROM giống hệt việc nạp cho PROM, tuy nhiên điểm khác căn bản là chúng cho phép xóa và nạp (lập trình) lại khi chưa chuẩn kết quả cuối cùng. Hình mạch điện 5.14 mô tả 1 mảng nhớ EPROM có khả năng nhớ 4 từ 4 bit có cấu tạo loại MOS kênh cảm ứng.

Hình 5.14b cho cấu tạo loại MOS có hai cực cửa (trong ký hiệu quy ước bổ sung thêm 1 đường đứt nét nằm song song với cực cửa thứ nhất truyền



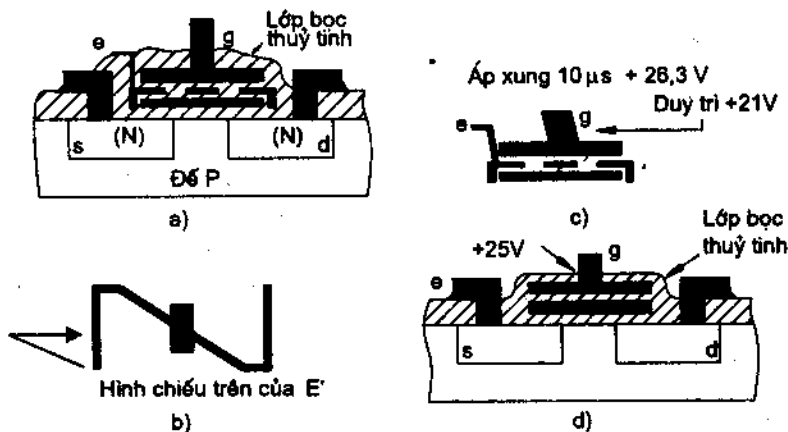
Hình 5.14. EPROM có cấu tạo từ MOS có cực cửa thả nổi (a) và cấu tạo có cực cửa thả nổi (b)

thống). Cực cửa này không nối với bên ngoài và được bao bọc kín trong một lớp thủy tinh (SiO_2) ở trạng thái thả nổi về điện :

* Khi cổng thả nổi không tích điện, MOS vận hành như loại bình thường (MOS kênh cảm ứng loại N) và cực cửa thứ nhất G_1 tiếp xúc với mạch ngoài điều khiển hướng dòng cực máng I_D khi có điện thế dương tác động.

* Khi đặt điện tích âm lên cổng thả nổi. MOS sẽ không hoạt động được vì làm mất khả năng điều khiển của G_1 do có màn ngăn hiện tượng cảm ứng tĩnh điện nên kênh cảm ứng giữa cực máng (D) và cực nguồn (S) không hình thành được, MOS đã bị loại ra khỏi mạch. Hình 5.14a là cấu tạo một mảng 4×4 ô nhớ loại EPROM này. Dấu (-) thể hiện cổng thả nổi G_2 đã được nạp điện áp âm và ô tương ứng đã được nhớ trạng thái 1, trường hợp ngược lại ô tương ứng lưu trạng thái 0. Mạch 5.14a đã được nạp dữ liệu $D_{03}D_{02}D_{01}D_{00} = 1011...$ $D_{33}D_{32}D_{31}D_{30} = 0100$. Khi có tín hiệu $WE_1 = 1$ cho phép đọc từ $D_{13}D_{12}D_{11}D_{10}$ ta nhận được kết quả tại đầu ra là 1000.

2. Xóa EPROM được thực hiện nhờ loại bỏ điện tử đã nạp cho cực G_2 thả nổi bằng cách phơi chip ra trước một tia cực tím rất mạnh qua một cửa sổ trong suốt đã chế tạo sẵn trên bề mặt các ô nhớ. Tia tím mang nhiều năng lượng để điện tử hấp thụ đủ vượt qua rào chắn thủy tinh mỏng làm mất điện



Hình 5.15. Mô tả sơ lược phương pháp xóa EEPROM (xóa bằng điện)

- a) Điện cực tại e được dùng để xóa ;
- b) Mặt trên của e ;
- c) Điện thế cần thiết trên e là $+26,3V$, $U_g = +21V$, phần e thường không nối để cách ly điện tích thả nổi hiệu quả hơn g ;
- d) Cấu trúc EEPROM xóa bằng cực cửa, chọn $U_g = +25V$.

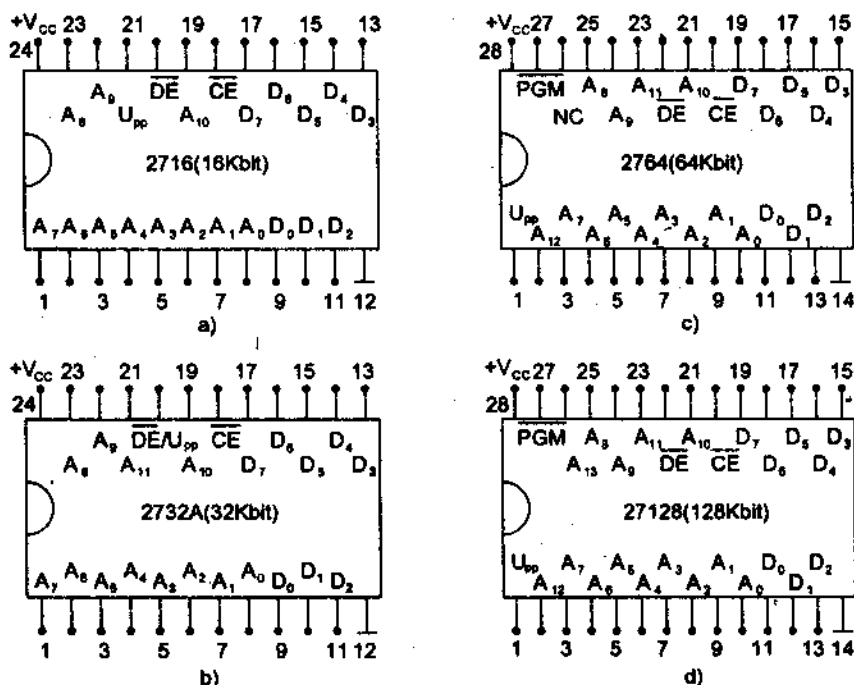
tích âm trên G_2 và do đó trả lại quyền điều khiển dòng I_D của G_1 như MOS cảm ứng thông thường. Do toàn bộ bề mặt chip bị tia tím chiếu vào, các thông tin (nạp bit 1) bị xóa sạch và phải nạp lại chip theo thông tin mới cần lưu giữ. Thời gian chiếu tia tím khoảng trên 15 phút, sau khi đã nạp xong có thể lấy giấy mờ dán che của sổ tuy rằng ánh sáng bình thường ít ảnh hưởng tới điện tích trên G_2 .

3. Có thể xóa EPROM bằng điện (EEPROM) khi đặt lên G_1 một điện thế dương khoảng 25V làm mất lớp điện tử đã bám ở G_2 thả nổi. Kỹ thuật này cho phép xóa có chọn lọc 1 từ hoặc vài từ... Hình 5.15 mô tả sơ lược phương pháp xóa EEPROM.

4. Khi nạp cho EPROM cần điện thế cao hơn mức chuẩn để nạp điện tích âm vào G_2 . Chọn $V_{GS} = 25V$, $V_{DS} = 20V$ sẽ đủ năng lượng cho điện tử vượt qua lớp SiO_2 tích tụ tại G_2 (các điện tử nóng) tạo nên một điện thế âm khóa MOS.

5.3.4. Một số chip IC thường gặp (hình 5.16)

1. EPROM xóa bằng tia tím (ví dụ 4 loại cho trên hình 5.16a, b, c, d).



Hình 5.16a,b,c,d

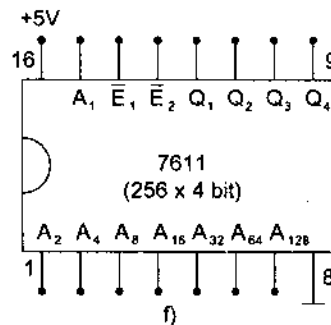
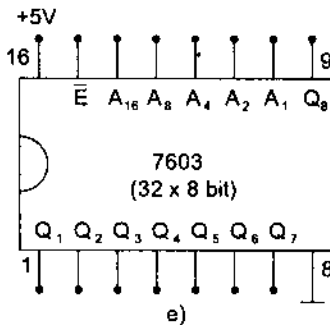
Với nhóm IC 2716/ 2732A/ 2764/ 2764/ 27128 chú ý là:

- Có chân U_{pp} để đặt điện áp cao khi ghi (nạp dữ liệu).
- Tương thích với TTL thường sử dụng trong hệ CPU.
- 27xx (x) 2 hay 3 số tiếp sau 27 chỉ dung lượng nhớ là ví dụ 2764 có dung lượng nhớ 64 Kbit.

2. Chip IC PROM được cho trên màn hình 5.16e, f với hai loại điển hình 7603 và 7611.

Với IC 7603 dung lượng 32 x 8 bit/16 chân tương thích TTL 5 đầu vào địa chỉ $A_1...A_{16}$, 8 đầu ra dữ liệu $Q_1...Q_8$, chế độ cho phép xuất khi $\bar{E} = 0$, khi $\bar{E} = 1$ các đầu $Q_1...Q_8$ bị thả nổi, nguồn +5V nạp dữ liệu bằng cách đốt cháy cầu chì loại đầu ra 3 trạng thái (với IC loại 7602 hở C).

Với IC 7611 dung lượng 256 x 4bit tương thích TTL, vào địa chỉ A_1 đến A_{128} đầu ra $Q_1Q_2Q_3Q_4$ loại 3 trạng thái khi làm việc thông thường (đọc) $\bar{E}_1 = 0$ $\bar{E}_2 = 0$, khi $\bar{E}_1 = 0$ $\bar{E}_2 = 1$ đầu ra thả nổi nạp bằng phương pháp đốt cháy cầu chì (với loại IC 7610 đầu ra hở cực colectơ).



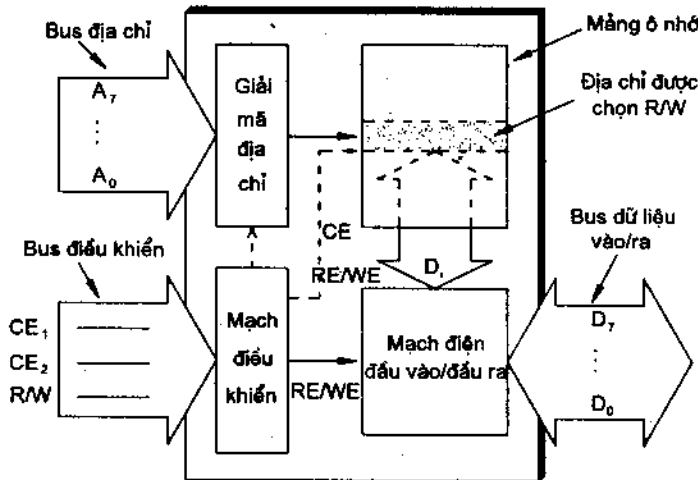
Hình 5.16e,f

5.4. BỘ NHỚ GHI – ĐỌC (Random Access Memory - viết tắt là RAM)

RAM là nhóm bộ nhớ quan trọng thứ hai bên cạnh ROM đã xét, có khả năng cho phép ghi lưu trữ dữ liệu thông tin tạm thời trong một thời gian nào đó, sau đó lại đọc thông tin đó ra để tiếp tục xử lý khi cần thiết nên nó còn được gọi tên là bộ nhớ loại ghi/đọc. Cần chú ý đến một đặc tính quan trọng khác của RAM là dữ liệu thông tin được ghi trong RAM có tính chất tạm trú, dễ bị xóa khi mất nguồn năng lượng cấp cho RAM nên cần để phòng trường hợp mất điện phải tự động cho RAM một nguồn nuôi phụ nhằm tránh hiện tượng trên khi quá trình xử lý còn chưa kết thúc, thông tin trong RAM còn có ích.

5.4.1. Cấu trúc khối của RAM

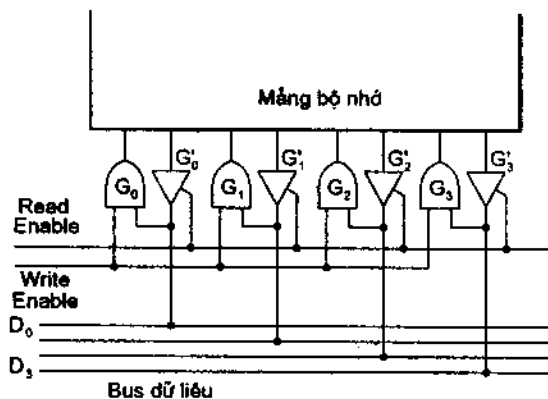
Cũng như ROM, RAM gồm 4 phần chính như thể hiện trên hình 5.17. Điểm khác biệt cần quan tâm là:



Hình 5.17. Cấu trúc 4 khối của 1 chip RAM có 8 bit dữ liệu và 8 bit địa chỉ

* Mạch điều khiển của RAM phải có thêm đầu vào R/W điều khiển hai quá trình cơ bản trong thao tác của RAM : ghi dữ liệu thông tin vào nó và xuất (đọc) thông tin đã ghi ra khỏi nó.

* Mạch đầu ra có khả năng kiểm soát hai chiều trước khi cho giao tiếp với kênh dữ liệu theo nguyên tắc lựa chọn : (đồng bộ với việc điều khiển R/W) đang đọc thì không ghi và ngược lại đang ghi thì không đọc, chế độ thứ 3 có thể là chờ quyết định.



Hình 5.18. Mạch vào ra 4 bit dữ liệu của 1 chip RAM : G_0, G_1, G_2, G_3 sẽ mở khi xung cho phép ghi $WE = 1$; G'_0, G'_1, G'_2, G'_3 sẽ mở khi xung cho phép đọc $RE = 1$

1. Mạch đầu vào/đầu ra của RAM phức tạp hơn của ROM do nhiệm vụ nó phải gánh vác. Hình 5.18 cho 1 dạng mạch vào/ ra với 4 bit dữ liệu D_0 đến D_3 .

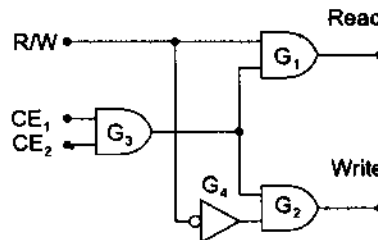
Các dữ liệu trên kênh dữ liệu sẽ qua G_0 , G_1 , G_2 và G_3 ghi vào bộ nhớ khi có tín hiệu cho phép ghi $WE = 1$. Còn khi có tín hiệu điều khiển cho phép đọc $RE = 1$ các cổng G_0 , G_1 , G_2 , G_3 được điều khiển mở (loại mạch 3 trạng thái) để xuất các dữ liệu từ bộ nhớ ra kênh dữ liệu, ta lưu ý rằng :

- Các thao tác trên phải đồng bộ với việc chọn xong địa chỉ ghi hay đọc trong bộ nhớ.

- Phần tử 3 trạng thái còn có khả năng đưa mức ra của $G_0' \dots G_3'$ lên mức trở kháng cao HiZ (chương 2) và do đó bộ nhớ được đặt trong tình trạng chờ: không ra cũng không vào dữ liệu mà tất cả các đầu ra dữ liệu được cách ly khỏi kênh dữ liệu cho an toàn.

2. Mạch điều khiển hình 5.19 là một dạng đơn giản đưa tới RAM 3 chế độ hoạt động khác nhau từ 3 đầu vào điều khiển CE_1 , CE_2 và R/W . Ở mỗi thời điểm chỉ một trong 3 chế độ được thiết lập:

- * Khi ở chế độ đọc, xung R/W ở mức 1 và chỉ đồng thời với các tín hiệu cho phép chọn $CE_1 = CE_2 = 1$ sẽ xuất hiện xung cho phép đọc (qua G_1) $RE = 1$ RAM được thiết lập chế độ đọc (chú ý liên hệ với mạch vào/ra hình 4.18). Đồng thời khi đó $\overline{R/W} = 0$ nên tín hiệu cho phép ghi $WE = 0$ (cấm ghi).



Hình 5.19. Mạch điều khiển 1 chip RAM đặt bộ nhớ vào 1 trong 3 chế độ: đọc/ghi/chờ

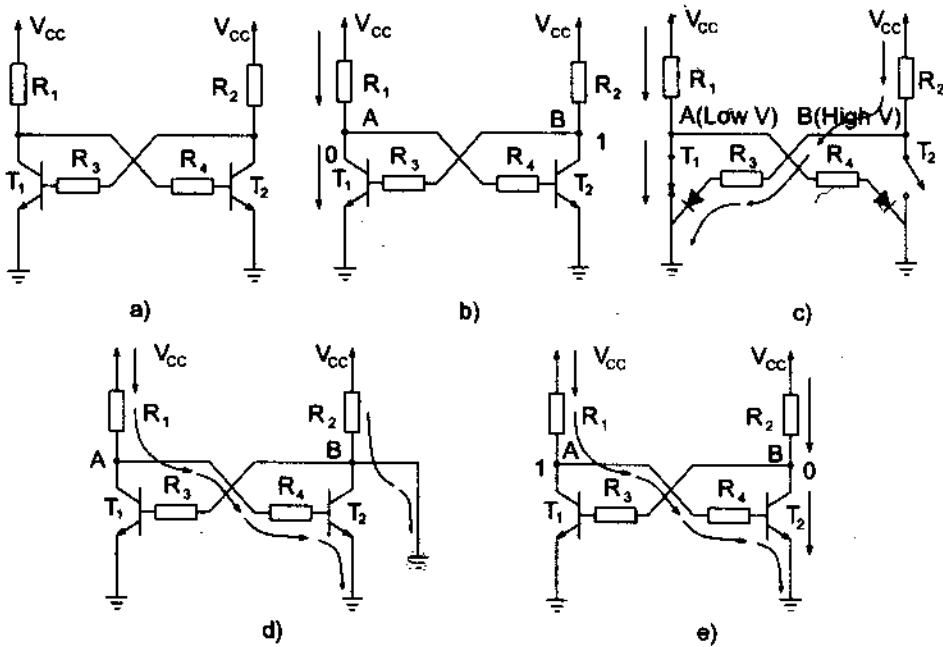
- * Khi ở chế độ ghi $R/W = 0$ qua G_4 , $\overline{R/W} = 1$ và cùng với lúc đó $CE_1 = CE_2 = 1$, G_2 mở xuất hiện $WE = 1$ cho phép ghi. Vì $R/W = 0$ nên G_1 cấm xung cho phép đọc ($RE = 0$).

- * Tín hiệu tích cực đồng thời $CE_1 = CE_2 = 1$ ở cả hai chế độ trên phải được chuyển đồng thời tới mảng ô nhớ nhằm thông báo việc xuất (khi đọc) hay việc nhập (khi ghi) tới địa chỉ ô nhớ đã được mạch giải mã chọn.

- * Khi $CE_1 \cdot CE_2 = 0$ (có ít nhất 1 tín hiệu CE không tích cực) mạch điều khiển ở hình 5.19 chuyển bộ nhớ sang chế độ chờ (STANDBY), bất chấp tín hiệu R/W tích cực hay không, lúc này $RE = 0$ và $WE = 0$.

5.4.2. Mảng ô nhớ dùng các tranzitor nhiều cực emito

Một số chip RAM sử dụng DFF NAND (hình 4.32b) làm ô nhớ có ưu điểm là tốc độ nạp xuất rất nhanh ($10^{-8}s$, $10^{-9}s$) tuy nhiên mỗi DFF chủ tố có thể chứa tới trên 16 tranzitor và hơn 6 điện trở. Với mảng ô nhớ dung lượng lớn thì đây là nhược điểm quan trọng vì mạch sẽ trở nên phức tạp và tiêu hao nhiều năng lượng. Một giải pháp tốt là mỗi ô nhớ chỉ sử dụng vài tranzitor mắc kiểu mạch đa hài lưỡng ổn (bistate multivibrator) có cấu trúc cơ bản dạng hình 5.20 và cấu trúc thực tế hình 5.21.



Hình 5.20 a) Mạch BM (đa hài lưỡng ổn) cơ bản, còn gọi là FF dùng tranzitor
b) và c) Trạng thái ổn định thứ nhất T_1^+, T_2^-
d) và e) Cho $V_B = 0$ kích hoạt, trạng thái ổn định thứ hai T_1^-, T_2^+

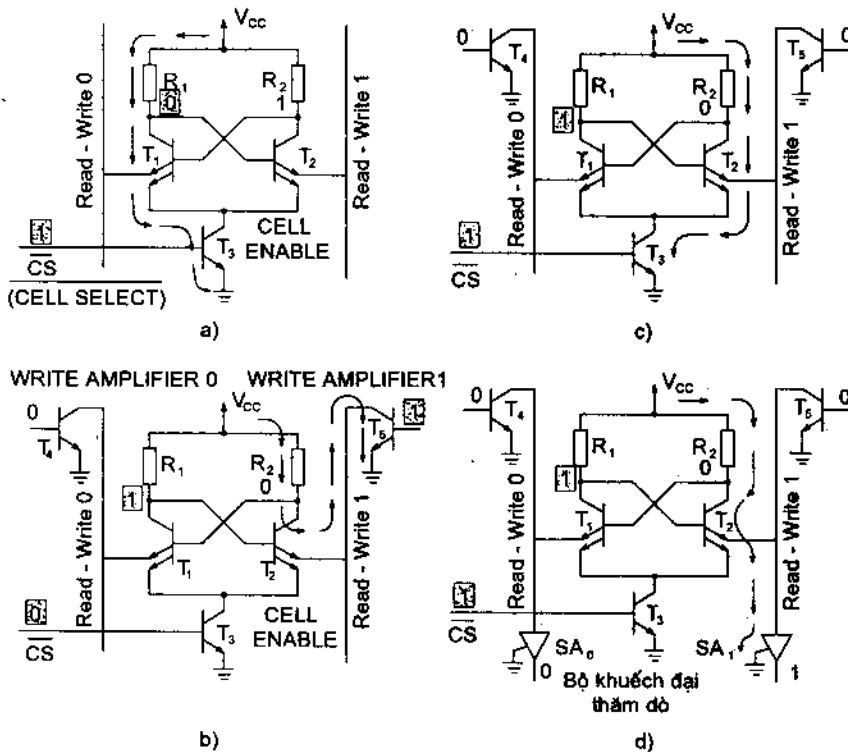
1. Ở mạch cơ bản hình 5.20a, tương tự như các Trigơ số, chỉ có hai trạng thái ổn định khi cấp nguồn $+V_{cc}$ là nếu T_1 nối mạch (dẫn dòng ký hiệu là T_1^+) thì T_2 ngắt mạch (không dẫn dòng, ký hiệu là T_2^-) và ngược lại T_1^-, T_2^+ . Trạng thái này chỉ thay đổi khi có tín hiệu ngoài áp vào hoặc mất V_{cc} .

* Khi T_1^+ , V_A ở mức thấp (0 logic), T_2^- , V_B ở mức cao (1 logic) như trên hình 5.20b. Mức cao V_B duy trì trạng thái T_1^+ (hình 5.20c).

* Ở hình 5.20d, nối điểm B tới mức 0V làm T_1^- điện thế tại A dâng lên mức cao, $V_A = 1$ làm T_2^+ .

* Khi ngắt đường nối 0V tại điểm B (hình 5.20e) mạch ổn định ở trạng thái $T_1^- T_2^+$ mức thế $V_A = 1$; $V_B = 0$, mức $V_A = 1$ duy trì cho T_2^+ chừng nào chưa có kích thích hay còn $+V_{CC}$. Việc nạp dữ liệu 1 vào vị trí điểm A đã được thực hiện.

2. Ở các mạch thực tế, để thực hiện 1 ô nhớ dạng hình 5.20a, T_1 và T_2 được thay thế bằng loại tranzitor nhiều emitter (hình 5.21), các đường nối tới các cực phát có chức năng điều khiển dòng điện qua tranzitor như nhau. Mỗi



Hình 5.21. 3 chế độ của 1 ô nhớ FF dùng BJT
Chế độ chờ (a,c), chế độ ghi (b) và chế độ đọc (d)

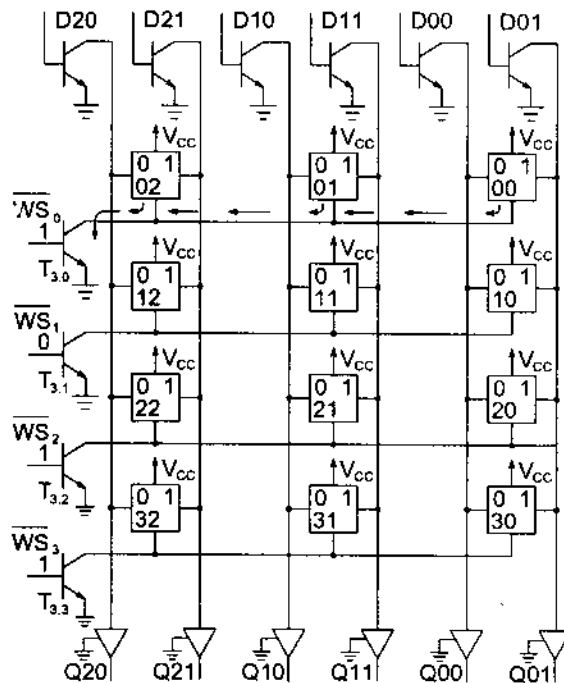
tranzitor có 1 emitor nối chung với đường tín hiệu điều khiển RE/WE để tạo nên mảng nhớ ma trận, trong đó 1 cặp $T_1^- T_2^-$ (loại nhiều emitor) cấu thành 1 ô nhớ của mảng có khả năng ở 3 chế độ khác nhau (chú ý ta quy ước giá trị của ô nhớ quy định ban đầu theo tranzitor dẫn điện).

* Khi tín hiệu điều khiển chọn ô nhớ $\overline{CS} = 1$ (hình 5.21a) T_3^+ dẫn điện (T_3^+) vì ô nhớ đang chứa bit 0 (T_1^+, T_2^-) nên mức 0 đang lưu được giữ lại: đây là chế độ chờ (đầu vào $\overline{CS}$ là ở mức tích cực thấp) chú ý rằng T_3^+ cùng với T_1^+ .

* Ở chế độ ghi (hình 5.21b) ví dụ muốn ghi bit 1 vào ô nhớ các tranzito thường T_4 và T_5 làm nhiệm vụ khuếch đại (xung) ghi, khi T_4^+ hoặc T_5^+ đường tín hiệu RW tương ứng sẽ nối 0V (mức logic 0). Xung $\overline{CS} = 0$ (làm T_3^-) xuất hiện đồng thời với $RW_1 = 0$ (do T_5^+ vì được mức thế cao đặt vào bazơ) đẩy T_2^+ và T_1^- , điện áp trên colector của T_1 dâng cao, mức 1 đã được lưu vào ô nhớ, sau đó mức $RW_1 = 1$ (T_5^-) và $\overline{CS} = 1$ làm T_3^+ (dẫn dòng cùng T_2^+). Kết thúc quá trình ghi, ô nhớ về chế độ chờ (hình 5.21c) với dữ liệu 1 đã được lưu.

* Khi ở chế độ đọc, mạch điện được bổ sung phân tử khuếch đại thăm dò (Sense Amplifier - gọi tắt là SA) có đặc điểm điện trở vào nhỏ và có một đầu nối 0V (xem hình 5.21d). Khi có dòng kích thích tới đầu vào của nó, mức ra của SA = 1. Vậy để đọc dữ liệu đang lưu, yêu cầu $\overline{CS} = 0$ (làm cho T_3^- , dòng của T_2^+ về qua nhánh emitor nối tới đường $RW_1 = 0$), điều

này làm SA_1 bị kích thích có dòng vào và lối ra có $SA_1 = 1$.



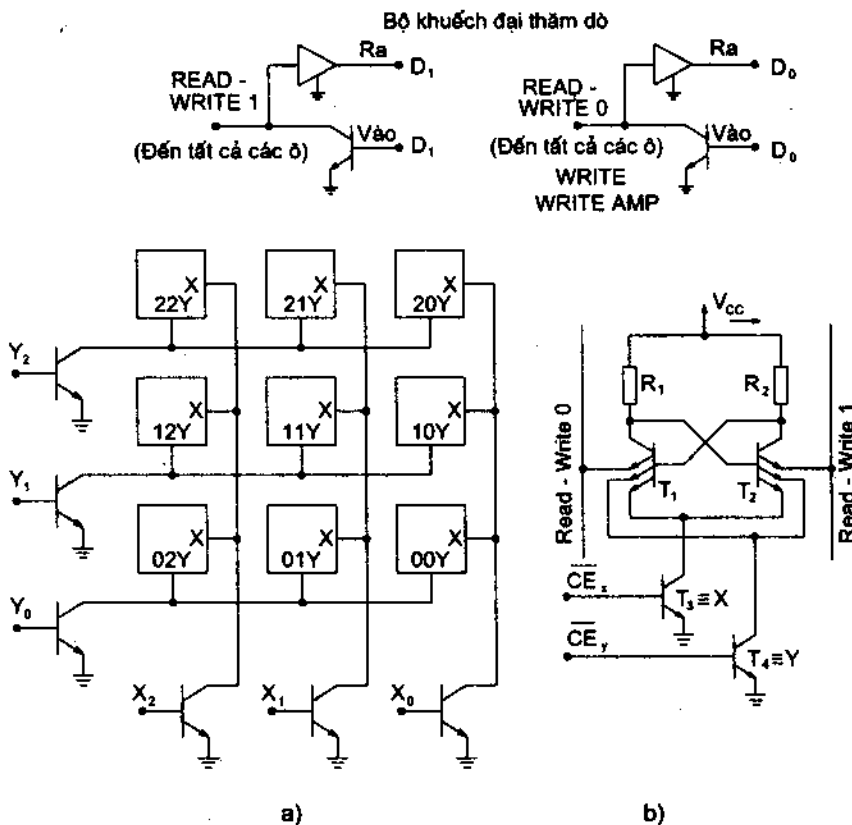
Hình 5.22. Mảng tuyến tính các ô nhớ loại 4x3 của một RAM với mỗi ô nhớ có dạng mạch hình 5.21

3. Hình 5.22 là mảng ô nhớ của các RAM tổ chức theo bảng tuyến tính (gọi là xếp theo từ mã lưu giữ, ở đây gồm có 4 từ 3 bit).

Đặc điểm của mạch (5.22) là các ô trong 1 từ (1 hàng) có chung tranzitor T_3 (hình 5.21) lúc đó đầu vào của T_3 sẽ là tín hiệu điều khiển chọn từ $\overline{WE_i}$ tương ứng ($i = 0, 1, 2, 3$).

* Ở chế độ chờ, dòng của các ô trong 1 từ đều qua T_{3i}^+ tương ứng (nếu đang chờ ở mức 0 thì dòng từ T_1^+ qua T_{3i}^+ còn ở mức 1 thì dòng từ T_2^+ qua T_{3i}^+).

* Để đọc/ghi, ra/vào một từ nào đó cụ thể (trong hình vẽ là từ ứng với các ô nhớ 12, 11 và 10), cần áp $\overline{WS_i} = 0$ vào T_{31} đẩy T_{31}^- , dòng sẽ rẽ tới các



Hình 5.23. a) Cấu trúc ma trận ô nhớ 9 x 1bit
b) Cấu tạo mạch điện 1 ô nhớ tranzitor có 3 cực emitơ

đây cột tương ứng ở mỗi ô, thủ tục đọc/ ghi được thực hiện qua mạch điện ra/vào đã xét ở hình 5.18 đồng bộ với điều khiển của R/W.

4. Hình 5.23a là cấu trúc mảng ma trận nhớ bit 9×1 gồm 9 ô nhớ dạng hình 5.23b.

Hình 5.23b mô tả cấu trúc 1 ô nhớ dùng các tranzitor có 3 cực phát (3 emitor), hai trong chúng được nối qua T_3 và T_4 tới tín hiệu điều khiển chọn (cho phép) ô CE ký hiệu là CE_X và CE_Y . Nếu có ít nhất một tín hiệu CE không tích cực thì ô nhớ ở chế độ chờ (vì T_3 hoặc T_4 hoặc cả hai dẫn điện, không có dòng chạy qua đầu đọc ghi RW_0 và RW_1). Chỉ khi CE_X và CE_Y đều tích cực thì ô nhớ mới được chọn tham gia vào thao tác đọc – ghi. Áp dụng điều trên vào mạch hình 5.23a với mỗi ô nhớ ví dụ với ô 11, nếu có tranzitor Y_1 hoặc X_1 hoặc cả hai dẫn điện thì ô 11 ở chế độ chờ, còn khi cả hai tranzitor Y_1 và X_1 đều không dẫn điện, ô 11 lên tích cực và hoạt động đọc – ghi sẽ xảy ra tại ô 11 (với một bit dữ liệu). Ở phía trên hình 5.23a có vẽ mạch ra lấy dữ liệu qua bộ khuếch đại thăm dò khi đọc hay đưa dữ liệu vào qua bộ khuếch đại ghi dùng một tranzitor khi ghi (có tính chất một bộ đảo). Hai đường đọc – ghi (Read – Write) chỉ số RW_1 , RW_0 sẽ đi vào tất cả các ô nhớ (ở đây không vẽ vào).

5.4.3. Mảng ô nhớ của SRAM dùng CMOS (SRAM gọi là RAM tĩnh)

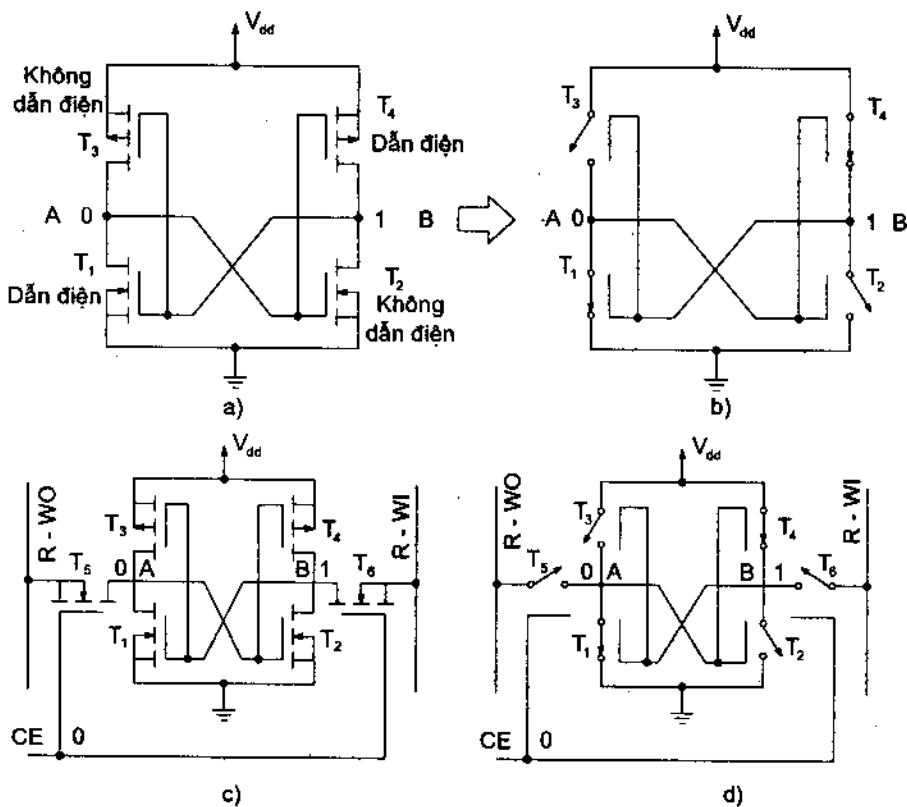
Ô nhớ dùng công nghệ CMOS được sử dụng phổ biến trong các bộ nhớ có dung lượng lớn vì ở chế độ lưu (tĩnh) chúng gần như không tiêu thụ năng lượng (công suất tiêu hao tại mỗi ô chỉ cỡ 10^{-9} đến 10^{-10} W) chỉ khi có thủ tục xuất/ nhập (đọc/ghi) xảy ra làm nó lật trạng thái (với mức trung bình là chỉ 50% số ô lật) mới tiêu hao một công suất cũng đủ nhỏ (thời gian chuyển ngắn) và công suất tiêu hao này sẽ tăng tỷ lệ với tần số lật của nó. Dùng CMOS tốc độ ghi/đọc (vào/ra) có xu hướng chậm hơn so với RAM dùng tranzitor lưỡng cực nhiều emitor.

1. Cấu trúc 1 ô nhớ dùng CMOS được cho trên hình 5.24a với 2 cặp MOSFET sinh đôi là T_1 (kênh N) với T_3 (kênh P) và T_2 (kênh N) với T_4 (kênh P) tất cả đều là kênh cảm ứng (loại kênh không có sẵn). Mạch có hai trạng thái ổn định giống một Flip Flop do tính chất đối xứng bù của nó:

Trạng thái thứ nhất là khi $T_1^+ T_3^-$ thì $T_2^- T_4^+$, khi đó điện thế tại cực máng của T_1 mức thấp ($V_A = 0$) của T_2 ở mức cao ($V_B = 1$). Còn khi $T_1^- T_3^+$

thời $T_2^+ T_4^-$ ta có $V_A = 1, V_B = 0$ đây là hai trạng thái lưu dữ liệu của ô nhớ (chờ). Các trạng thái đã nêu trên là ổn định bền vững nếu còn duy trì $+V_{dd}$ và không có tín hiệu điều khiển thích hợp tác động vào ô nhớ. Ví dụ, khi $T_1^+ T_3^-$ và $T_2^- T_4^+$ luôn có điện thế V_B dương duy trì cho T_1^+ (hay ngược lại có điện thế V_A dương mức cao duy trì cho T_2^+ ở trạng thái ổn định thứ hai).

2. Hình 5.24c ngoài $T_1 T_2 T_3 T_4$ còn bổ sung T_5 và T_6 nối trực tiếp với các đường ghi đọc RW_0 và RW_1 , T_5 và T_6 chịu sự điều khiển trực tiếp của tín hiệu chọn ô nhớ CE , khi $CE = 0$, T_5 và T_6 ngắt mạch (T_5^-, T_6^-) ô nhớ

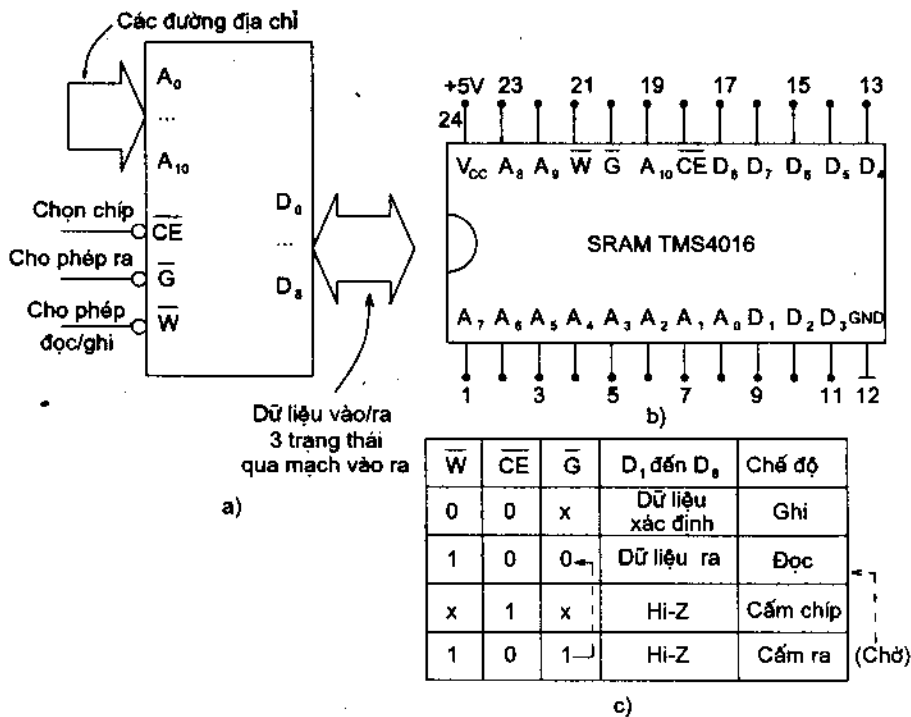


Hình 5.24. a) Một ô nhớ CMOS dùng 2 cặp FET bù $T_1 T_3$ và $T_2 T_4$

b) Trạng thái lưu dữ liệu của ô nhớ $V_B = 1, V_A = 0$

c) Ô nhớ có đầu vào giao tiếp với tín hiệu cho phép chọn CE thông qua T_5 và T_6 tới đường RW_1

d) Trạng thái chờ, ô nhớ cách ly hoàn toàn với các đường RW

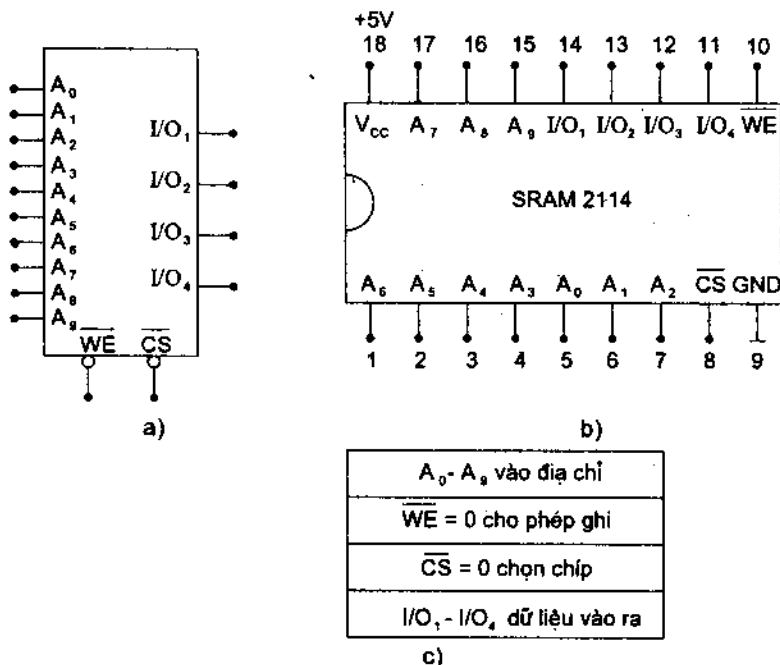


Hình 5.25. SRAM TMS 4016 2048 x 8 bit
a) Sơ đồ chức năng để phân tích, tính dung lượng nhớ ;
b) Bố trí chân cần thiết cho thiết kế lắp ráp ;
c) Bảng chân lý (cần thiết khi chọn chế độ sử dụng).

hoàn toàn cách ly khỏi các đường dữ liệu RW (hình 5.24d). Khi $CE = 1$, ô nhớ được chọn cho phép ghi/đọc. T_5 và T_6 dẫn điện tuy nhiên chỉ những đường có tín hiệu $\overline{RW} = 0$ thì mức 1 mới qua FET (trong ví dụ trên qua T_6) tác động tới phân tử khuếch đại thăm dò gây ra mức 1 ở đầu ra (dữ liệu 1 được đọc). Còn khi ghi vào cũng thông qua xung chọn $CE = 1$ với đường ví dụ $\overline{RW}_0 = 1$; thông qua T_5^+ (nhờ $CE = 1$) tới điểm A kích thích T_2^+ và do đó dập tắt T_1^- , như vậy đã ghi được mức 1 vào ô nhớ ($V_A = 1$), mức nhớ nằm lâu chừng nào còn V_{cc} .

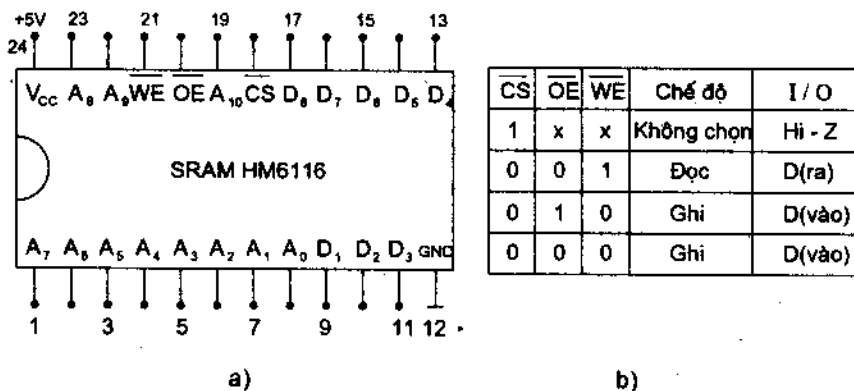
3. Hình 5.25 giới thiệu sơ đồ chức năng (a) phân bố chân (b) và bảng chân lý (c) của chip ICS RAM loại TMS 4016 có dung lượng 2048 x 8 bit.

Hình 5.26 giới thiệu SRAM 2114-2 (INTEL) dung lượng 1024 từ x 4bit tốc độ thâm nhập 200ns, công suất 525mW.



Hình 5.26. a) Sơ đồ logic ; b) Bố trí chân ; c) Bảng chế độ của SRAM 2114

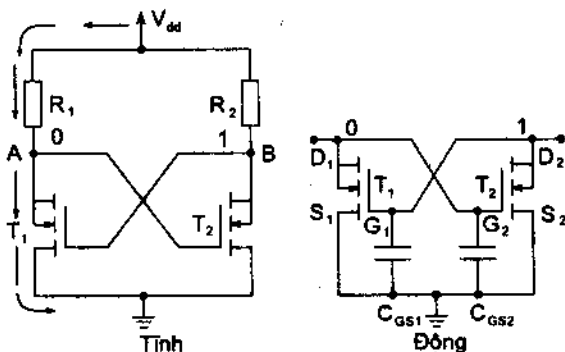
Hình 5.27 giới thiệu bố trí chân và bảng chân lý của SRAM CMOS HM 6116 tốc độ nhanh 2048 từ x 8bit. Thời gian thâm nhập 120ns, P(chờ) = 100 μ W, P(hoạt động) = 180mW.



Hình 5.27. a) Bố trí chân SRAM HM 6116 ; b) Bảng chân lý của HM 6116.

5.4.4. Bộ nhớ RAM động (DRAM) dùng CMOS

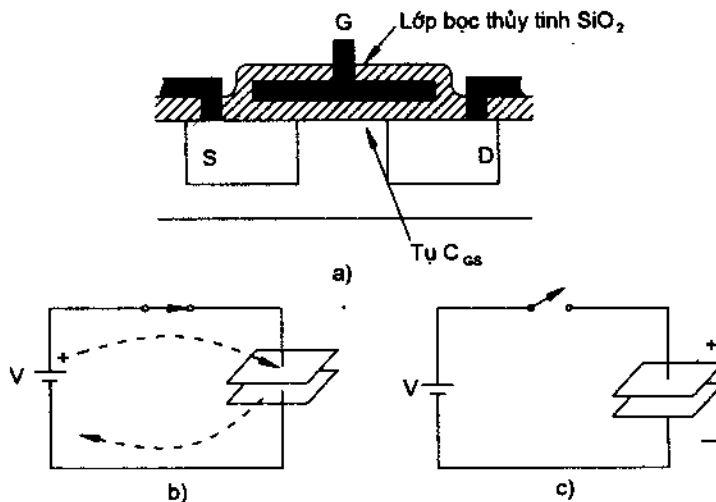
1. Với trở kháng đầu vào của MOS rất lớn, có thể chế tạo loại ô nhớ sử dụng sự có mặt hay không của các điện tích trên cực cửa của MOSFET để lưu giữ thông tin trong một khoảng thời gian nhất định, bộ nhớ với các ô nhớ dạng này gọi là RAM động hay DRAM. Hình 5.28 mô tả 1 đơn vị nhớ (ô nhớ 1 bit) của DRAM dùng MOSFET sử dụng tụ điện C_{GS} của T_1 hoặc T_2 để nhớ điện tích (dữ liệu). Với cách nhớ dùng tụ điện, theo thời gian bộ nhớ này có xu hướng mất thông tin



Hình 5.28. Cấu trúc ô nhớ cơ bản DRAM (b) phân biệt với 1 ô nhớ SRAM (a)

do dòng điện dò của tụ gây ra và vì thế theo định kỳ phải nạp lại cho C_{GS} (làm tươi bộ nhớ) để tránh làm mất (chống xu hướng quên) thông tin đã lưu.

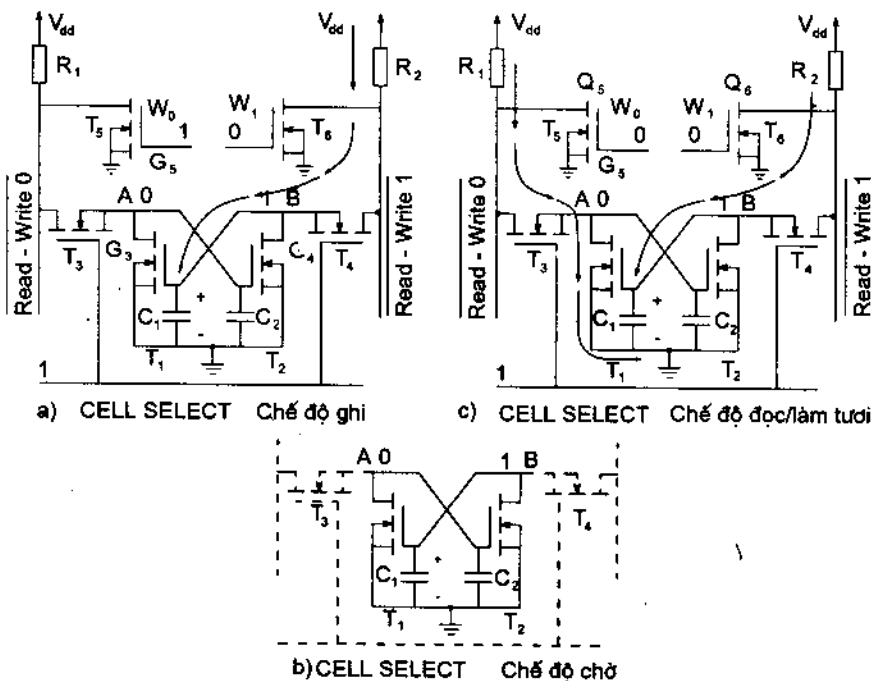
Hình 5.28 thể hiện 2 ô nhớ của SRAM (a) và DRAM (b). Ở ô nhớ tĩnh các tranzitor T_1 và T_2 được nối qua R_1 và R_2 (hay T_3 và T_4) tới nguồn



Hình 5.29. a) Tụ điện C_{GS} hình thành giữa lớp kim loại làm cực G với kênh dẫn điện bán dẫn qua lớp điện môi thủy tinh SiO_2 ; b) Dùng nguồn V nạp cho tụ; c) Điện tích nhớ lưu lại trên tụ khi ngắt nguồn nạp.

$+V_{DD}$, muốn lưu lại thông tin này ($T_1^+ T_2^-$, $V_A = 0$, $V_B = 1$) ô nhớ phải duy trì $+V_{DD}$. Ô nhớ của DRAM không nối trực tiếp tới V_{DD} , tuy nhiên mạch có thêm 2 linh kiện là C_{GS1} và C_{GS2} , đây là các tụ điện hình thành tự nhiên do bản chất cấu tạo của MOSFET (hình 5.29a) và quá trình nạp điện tích nhớ cho tụ minh họa trên hình 5.29b và c. Do hai bản tụ C_{GS} không được cách điện tuyệt đối nên sẽ có dòng điện dò (chỉ $10^{-9}A$) làm điện tích nhớ trên C_{GS} bị hao hụt đi theo thời gian có nguy cơ làm mất thông tin (khi mức điện thế trên tụ trở nên thấp hơn mức cao cho phép sẽ mất dữ liệu 1) vấn đề nạp lại điện tích cho tụ (làm tươi ô nhớ) trở thành vấn đề quan trọng để chống mất thông tin được lưu trữ trong ô nhớ.

2. Hoạt động của ô nhớ của RAM được làm rõ ở hình 5.30 với 3 chế độ tương ứng : chế độ ghi, chế độ chờ và chế độ đọc kết hợp làm tươi ô nhớ. Ô nhớ cơ bản gồm $T_1 T_2$ và $C_1 C_2$, các đường RW_0 và RW_1 được đưa vào ô nhớ qua T_3 và T_4 và đường CS (cellselect - chọn ô) tác động tới G_3 và G_4 .



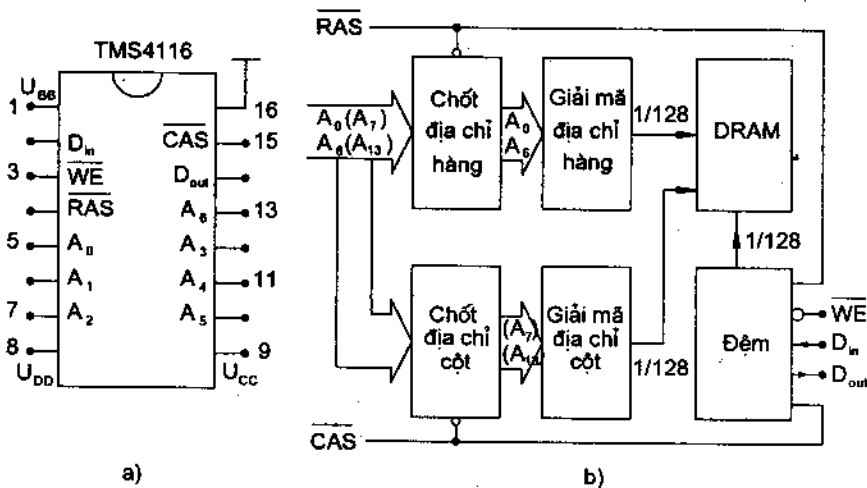
Hình 5.30. Các chế độ của DRAM

* Ở chế độ ghi (hình 5.30a) đang có bit 0 được ghi vào ô nhớ (xác định theo T_1^+) khi đó $CS = 1$, $T_3^+ T_4^+$ nối ô nhớ tới đường RW_0 và RW_1 . Nếu lúc này RW_0 ở tích cực (mức thấp, $\overline{RW_0} = 0$ bằng cách áp $W_0 = 1$ vào cực G_5 để RW_0 nối đất lúc đó $W_1 = 0$ nên T_6^-). Điểm A nối 0V làm T_2^- điện thế dương tại đường RW_1 qua T_4^+ xuất hiện tại điểm B tụ C_1 được nạp tới điện áp V_{DD} làm T_1^+ ; $V_A = 0$ (qua T_3 tới đường RW_0).

* Chế độ chờ CS trở về 0, T_3^-, T_4^- cô lập ô nhớ. T_1^+ duy trì điều kiện A nối đất ($V_A = 0$) do V trên C_1 tiếp tục dẫn nối tắt 2 cực của C_2 tới đất: T_2^- mạch ổn định (chờ) với T_1^+ và T_2^- $V_A = 0$, $V_B = 1$.

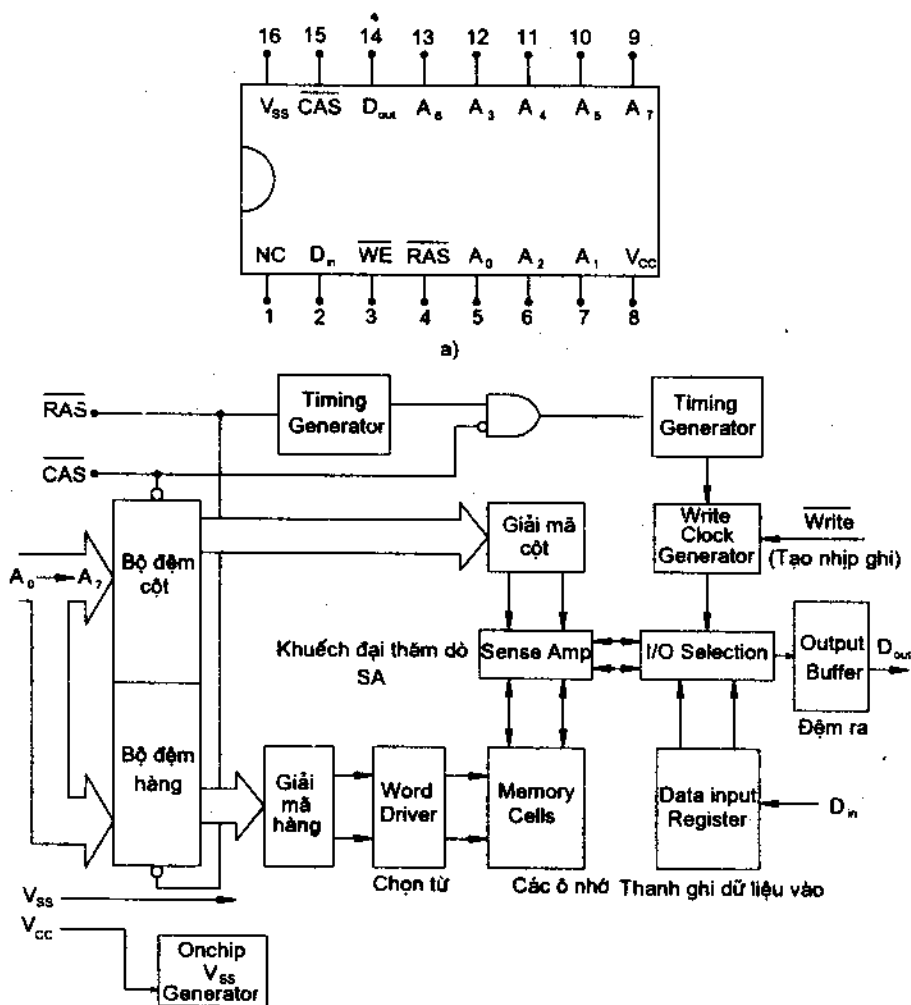
* Hình 5.30c mô tả quá trình đọc, khi đó $CS = 1$ (ô nhớ được chọn lên tích cực chuẩn bị xuất dữ liệu) làm $T_3^+ T_4^+$, T_1 đang dẫn nếu $RW_0 = 0$, bit 0 hình thành trên đường RW_0 cho biết đã đọc dữ liệu 0 ra khỏi ô nhớ. Nếu bit nhớ là 1, $RW_1 = 0$.

* Trong thời gian chờ (hình 5.30b) điện tích trên C_1 (hoặc C_2) bị hao hụt cần nạp lại, khi $W_0 = W_1 = 0$ thì $RW_0 = RW_1 = 1$ qua T_3^+ hay T_4^+ nạp cho C_2 hay C_1 từ nguồn $+V_{DD}$ (xem chiều dòng điện nạp cho C_1 từ RW_1 và



Hình 5.31

C_2 từ RW_0). Như vậy hoạt động ở chế độ đọc ô nhớ được kết hợp làm tươi. Tụ C_1 C_2 trong 1 ô nhớ chỉ có điện dung cỡ 5.10^{-14} F nên nhanh bị mất điện tích, thường trong DRAM cỡ 2.10^{-3} s nó lại được làm tươi một lần bằng 1 chu kỳ làm tươi định sẵn không phụ thuộc vào chế độ đọc (hay cứ 2ms DRAM lại được đọc 1 lần) mỗi lần kéo dài cỡ 4.10^{-5} s, điều này làm cho



Hình 5.32. DRAM MN 4164 65Kbit

a) Bố trí chân;

b) Sơ đồ cấu trúc của DRAM MN 6164 loại 65Kbit (panasonic).

việc sử dụng rắc rối hơn và tín hiệu ra thường không phù hợp với TTL. Khi cần tăng dung lượng của RAM nói chung có thể ghép song song nhiều chip RAM có dung lượng nhỏ và kết hợp nhiều nhánh song song qua một mạch giải mã địa chỉ chọn nhánh.

3. Hình 5.31a và b là sơ đồ bố trí chân và sơ đồ cấu trúc của DRAM.

TMS 4116 loại MOS dung lượng 16Kx1bit có tổng cộng $2^{14} = 16384$ ô nhớ, số đường địa chỉ yêu cầu là 14. Thực ra theo phương pháp chọn kênh chỉ cần dùng 7 đường chứa thông tin về hàng, sau đó lại chứa thông tin về cột. Muốn vậy người ta sử dụng hai tín hiệu $\overline{RAS}$ và $\overline{CAS}$ tương ứng: khi $\overline{RAS} = 0$, thông tin trên các đường địa chỉ sẽ được mở qua mạch chốt địa chỉ hàng và khi $\overline{CAS} = 0$ mở mạch chốt địa chỉ cột. Cắm trạng thái $\overline{RAS} = \overline{CAS} = 0$ (làm rối loạn việc xác định địa chỉ ô nhớ).

Khi đường có giá trị $\overline{WE} = 0$ là cho phép ghi để thông tin D_{in} được nhập vào địa chỉ đã chọn. Khi đọc, $\overline{WE} = 1$ từ địa chỉ ô nhớ được chọn, thông tin sẽ được xuất ra đường D_{out} .

Hình 5.32a và b là bố trí chân và sơ đồ cấu trúc của DRAM MN4164 loại NMOS 65Kbit. Số ô nhớ : 65536×1 , tốc độ truy cập 150ns, 128 chu kỳ làm tươi/2ms. Công suất ở chế độ tích cực 275mW, ở chế độ chờ 27,5mW.

5.5. KHÁI NIỆM VỀ BUS

5.5.1. Trong một hệ thống gồm nhiều khối thiết bị số, các khối này phải trao đổi thông tin cho nhau. Thông tin trao đổi có 3 dạng:

- Thông tin chứa nội dung cần trao đổi gọi là thông tin dữ liệu.
- Thông tin chứa địa chỉ nguồn và đích cần giao tiếp, kết nối nhau gọi là thông tin địa chỉ.
- Thông tin để điều khiển nhịp nhàng việc phối hợp thao tác (đồng bộ với nhau) gọi là thông tin điều khiển.

Các đường truyền dẫn nếu nối trực tiếp từ thiết bị này sang thiết bị khác sẽ trở nên quá phức tạp nếu hệ thống bao gồm nhiều thiết bị (điều này lại thường gặp trong thực tế). Biện pháp có hiệu quả là sử dụng một hệ thống tập hợp các dây dẫn chung gọi là kênh hay Bus. Tất cả các đầu vào và đầu ra của mỗi thiết bị số được giao tiếp với kênh chung này. Nhờ có các tín hiệu

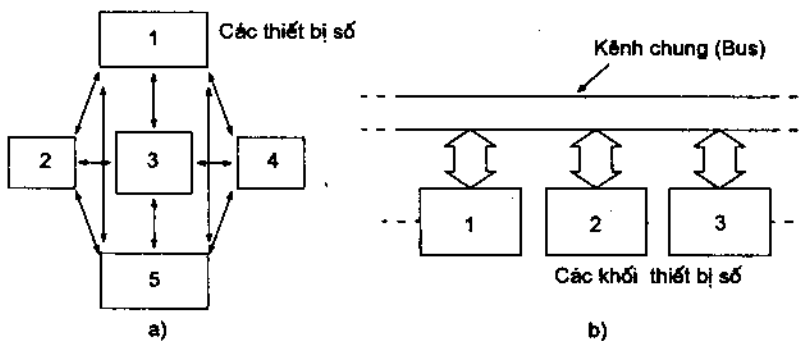
điều khiển mà tại một thời điểm xác định chỉ có một vài thiết bị có liên quan tới hoạt động đang diễn ra được phép giao tiếp với Bus, những thiết bị không liên quan trong lúc đó sẽ bị tách rời khỏi Bus. Như vậy cấu trúc Bus có thể mô tả trên hình 5.33b (để so sánh có hình 5.33a là cách nối trực tiếp không dùng Bus), với 3 dạng thông tin cần trao đổi, hệ thống Bus tương ứng gồm 3 nhóm : Bus dữ liệu, Bus địa chỉ và Bus điều khiển.

Có thể vẽ quy ước 1 dây (hoặc một nét to) tượng trưng cho Bus và ghi số lượng đường dây (số bit thông tin) trên nó.

5.5.2. Việc nối các đầu vào/đầu ra của các khối số tới một kênh chung nảy sinh một số vấn đề cần phải giải quyết là :

1. Tranh chấp cao/thấp trên 1 dây dẫn chung khi ở một đầu ra đang có kéo lên mức 1 còn tại 1 đầu ra khác đang có kéo dây xuống mức 0 và thường mức 0 sẽ thắng thế (phần 2.1 chương 2), hệ quả là phần tử này sẽ nối tắt phần tử kia.

2. Một đầu ra phải cung cấp cho quá nhiều đầu vào tải mà năng lực phun dòng hay hút dòng của nó lại có hạn sẽ gây quá tải làm hỏng mạch điện tăng ra.



Hình 5.33. a) Nối dây kiểu trực tiếp ; b) Dùng kênh chung (Bus)

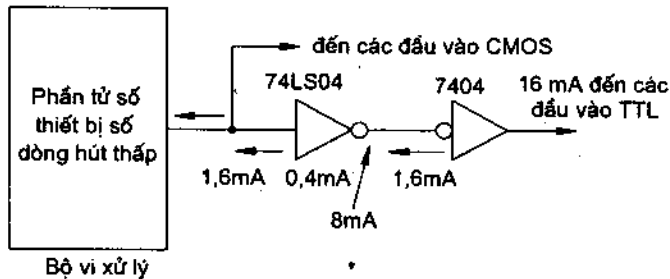
3. Các đầu vào tải yêu cầu kích thích không đồng nhất phát sinh gây quá tải đầu vào (với phần tử nhập năng lượng nhiều hơn) hay không đủ kích thích với phần tử được tiếp nhận ít quá.

5.5.3. Các rắc rối trên có thể được giải quyết nhờ các biện pháp sau:

1. Sử dụng các bộ đệm 3 trạng thái ở tăng ra để có trạng thái HiZ (trở kháng cao) khi nó không được chọn làm việc (bị cách ly).

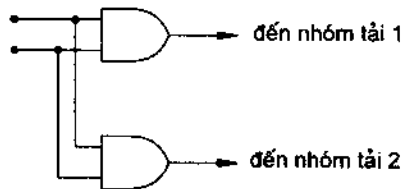
2. Sử dụng các phần tử logic hờ cực góp ở đầu ra để khi chúng cùng được chọn làm việc sẽ điều tiết được hệ số tải cho chúng nhờ tải mắc từ ngoài và nhờ khả năng dòng lớn của tăng ra (chương 2, phần 2.11).

3. Sử dụng các phân tử có năng lượng đầu ra cao kích thích các phân tử tải đòi hỏi năng lượng vào thấp hoặc rất thấp (ví dụ 1 cổng TTL đầu ra có thể kích thích 75 đầu vào CMOS) hoặc nâng hệ số tải của một phân tử bằng cách dùng phân tử đệm có hệ số tải cao hơn trước khi kích thích tải (hình 5.34).



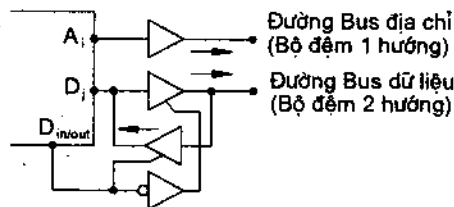
Hình 5.34. Dùng phân tử đệm nâng hệ số tải

4. Sử dụng 2 hoặc nhiều nhánh kích thích song song dùng 2 hoặc nhiều phân tử giống hệt nhau đầu song song như thể hiện trên hình 5.35.



Hình 5.35. Ghép song song 2 nhánh để nâng hệ số tải

5. Sử dụng các bộ đệm / kích thích 1 chiều nếu hướng thông tin từ phân tử tới Bus chỉ đi theo 1 chiều (chiều cho hay chiều nhận) hoặc dùng phân tử đệm / kích thích 2 chiều khi cần trao đổi thông tin 2 chiều kiểu cho / nhận (xem hình 5.36).

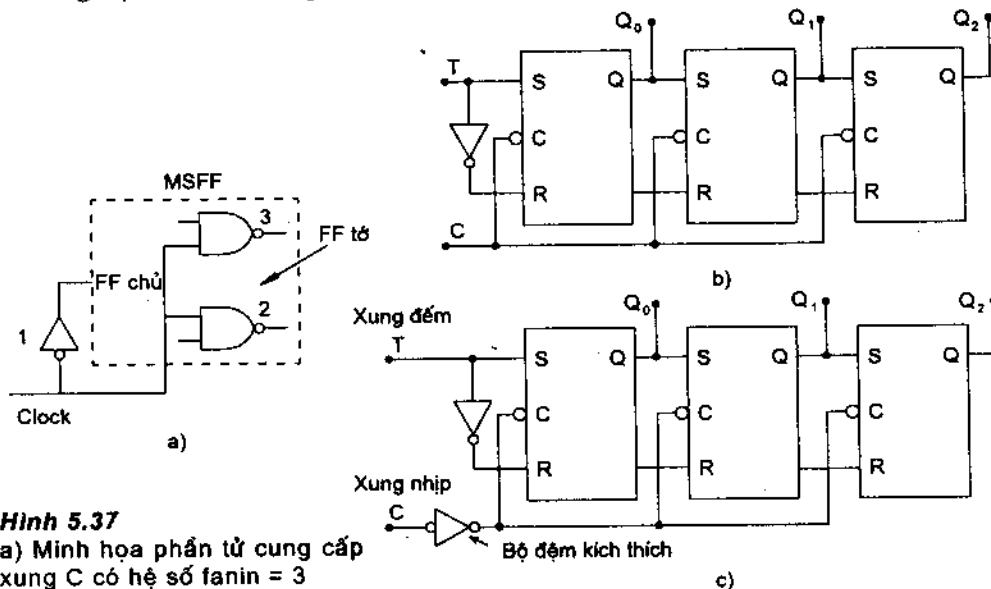


Hình 5.36. Dùng bộ đệm một hướng hay hai hướng để định hướng luồng thông tin giao tiếp với Bus.

6. Hạ thấp hệ số fanin bằng các phân tử đệm (xem hình 5.37) hệ số fanout cho khả năng kích thích tải của một đầu ra, hệ số fanin mô tả cho một đầu vào đặt trên 1 đầu ra. Xét mạch hình 5.37a: phân tử cung cấp xung nhịp phải kích thích 3 đầu vào do đó ta xác định được nguồn xung C có hệ số fanin là 3. Với mạch hình 5.37b là một bộ đếm

hệ 8 có hệ số fanin của C là 9 vì phải dùng 3 FF dạng hình 5.37a.

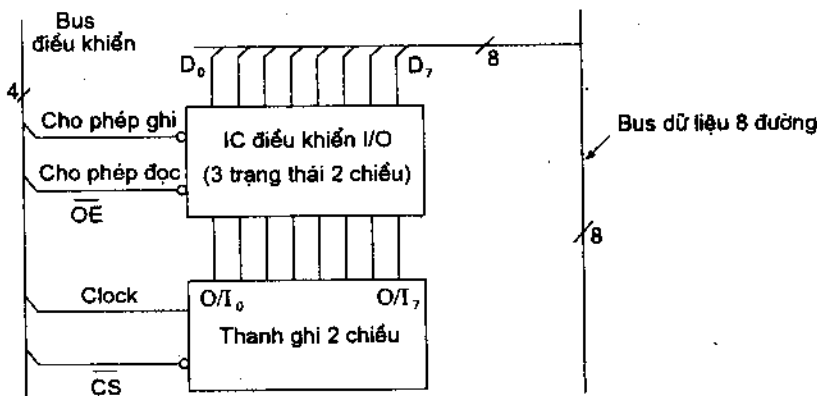
Hình 5.37c sử dụng thêm một bộ đếm kích thích cho bộ đếm hình 5.37b rõ ràng hệ số fanin của phần tử tạo xung nhịp C bằng 1.



Hình 5.37

- a) Minh họa phần tử cung cấp xung C có hệ số fanin = 3
- b) Bộ đếm hệ 8 với hệ số fanin của C là 9
- c) Thêm bộ đếm kích thích không đảo hệ số fanin của C bằng 1

5.5.4. Hoạt động truyền dữ liệu trên Bus được thực hiện theo phương thức song song từng nhóm 4, 8, 16, 32 bit dữ liệu với yêu cầu phối hợp: đầu ra của một thiết bị (ví dụ thanh ghi của nó) được cho phép để đặt dữ liệu



Hình 5.38. Gép IC thanh ghi dữ liệu 2 chiều với Bus dữ liệu qua IC điều khiển Bus

(xuất) lên Bus, đầu vào của một thiết bị khác (ví dụ thanh chốt của nó) được cho phép để tiếp nhận dữ liệu này từ Bus và khoá chúng vào mạch bên trong. Các hoạt động này được thực hiện theo nhịp điều khiển chung được thiết kế để đảm bảo quá trình gửi và nhận chính xác, kịp thời. Các tín hiệu cho phép được truyền trên Bus điều khiển thông qua việc kết hợp với Bus địa chỉ để tìm kiếm định vị các phần tử được cho phép tham gia vào hoạt động gửi / nhận đang cần thực hiện. Trước khi xuất nhập dữ liệu ra Bus mọi thiết bị phải được ghép nối qua một giao diện gọi là IC điều khiển Bus là loại có đầu ra 3 trạng thái phù hợp với yêu cầu 1 hướng hay 2 hướng tùy theo tính chất của thiết bị : ví dụ với ROM hướng tín hiệu điều khiển là đi vào còn hướng dữ liệu là đi ra Bus, hướng dữ liệu của RAM lại là hai chiều như đã xét ở tiết trên. Ví dụ trên hình 5.38 minh họa ký hiệu thu gọn của 1 Bus dữ liệu 8 đường cho 1 thanh ghi 2 chiều. Nhiều thiết bị có bố trí phần mạch ra 3 trạng thái nằm ngay trong vỏ IC.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 5

5.1. Bộ nhớ có nhiệm vụ gì ?

Hãy nêu các đầu vào, đầu ra tối thiểu của 1 bộ nhớ bán dẫn.

Tại sao có thể dùng Flop Flop làm 1 ô nhớ được ?

Các thủ tục tối thiểu để một bộ nhớ có thể hoạt động được là gì ?

5.2. Phân loại bộ nhớ theo chức năng:

a) Hãy phân biệt các thuộc tính của ROM và của RAM.

b) Dung lượng của một bộ nhớ là gì ? Cách tính dung lượng của bộ nhớ dựa trên những đặc điểm nào ? cho ví dụ minh họa ?

c) Nói rằng 1 bộ nhớ có dung lượng là 64 Kbyte và 1 bộ nhớ khác có dung lượng là 512 Kbit có gì khác nhau? Cần dùng bao nhiêu đường địa chỉ $A_0 \dots A_k$ cho mỗi loại này ?

d) Bộ nhớ chính và bộ nhớ phụ khác nhau ở những điểm nào ?

5.3. Tại sao lại muốn tăng dung lượng của bộ nhớ ?

Bằng cách nào có thể làm tăng dung lượng của bộ nhớ ?

Thời gian thâm nhập bộ nhớ là gì ?

Có mối liên hệ gì giữa thời gian thâm nhập bộ nhớ và dung lượng của nó ?

5.4. a) Hãy nêu các thuộc tính cơ bản của ROM.

b) Hãy vẽ 1 cấu trúc chung của ROM và giải thích tóm tắt nhiệm vụ của các khối chức năng có trong ROM.

c) Khối mạch điều khiển có những nhiệm vụ gì ? Nhiệm vụ nào là quan trọng nhất.

5.5. Hãy phân biệt cấu trúc các ô nhớ trong mảng bộ nhớ.

a) Loại cấu trúc tuyến tính có đặc điểm gì ? Nhược điểm lớn nhất của cấu trúc này ?

b) Cấu trúc mảng kiểu ma trận được tổ chức như thế nào ?

Một ma trận có N_1 hàng, N_2 cột và N_3 lớp ghép song song các mảng sẽ có tổng cộng bao nhiêu ô nhớ ?

5.6. Hãy vẽ một ma trận các ô nhớ bao gồm 4 hàng và 4 cột. Cần bao nhiêu bit đánh địa chỉ cho các ô nhớ ?

Khi nâng lên thành ma trận 64 hàng và 8 cột thì sẽ cần bao nhiêu bit địa chỉ để gán cho từng ô nhớ của nó ?

5.7. Phân biệt và so sánh hai loại giải mã địa chỉ cho mảng bộ nhớ tuyến tính và cho mảng bộ nhớ dạng ma trận vẽ:

a) Cấu tạo có đặc điểm gì khác nhau ?

b) Với bộ nhớ dung lượng lớn dùng loại nào có ưu điểm hơn ?

c) Hãy so sánh trong ví dụ cấu trúc ma trận 4×4 và cấu trúc mảng tuyến tính 4×4 của bộ nhớ.

5.8. Mạch điện đầu ra của ROM có đặc điểm gì ? hãy nêu nhiệm vụ của nó.

Tại sao phải dùng phần tử 3 trạng thái để thực hiện mạch điện đầu ra ?

Vai trò của mạch điều khiển đối với mạch ra ?

5.9. a) ROM mặt nạ dùng điốt nối được thực hiện như thế nào ? đặc điểm chính của loại này là gì khi tiến hành nạp (lập trình) cho ROM.

b) Khi dùng BJT (tranzitor lưỡng cực) hay MOSFET thay thế cho các điốt trong ROM mặt nạ cần lưu ý cách thức nạp dữ liệu vào ROM được thực hiện như thế nào ?

5.10. ROM có cấu chỉ là gì ? Việc nạp dữ liệu 1 hay 0 vào một ô nhớ của loại này được thực hiện như thế nào ? PROM và ROM có cấu chỉ có phải là 2 tên gọi khác nhau của một loại ROM không ? PROM có ưu nhược điểm gì ?

5.11.a) EPROM là gì ? Tại sao nói rằng EPROM có nhiều ưu điểm hơn so với PROM hay ROM mặt nạ ?

b) Phương pháp nạp cho EPROM có đặc điểm gì ?

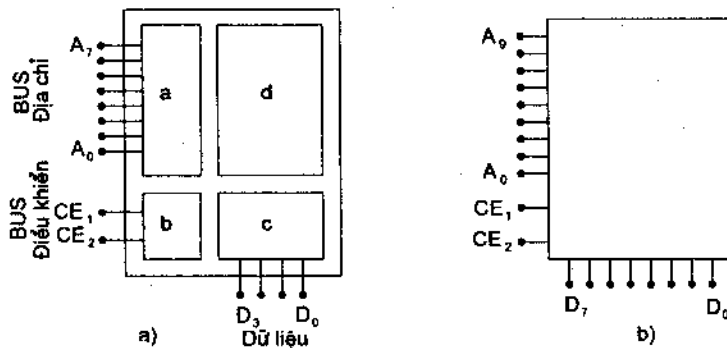
c) Khi xóa EPROM cần làm những việc gì ? tại sao phải che cửa sổ ?

d) Việc nạp hay xóa cho EPROM bằng điện được làm như thế nào đối với một ô nhớ dùng tranzitor trường theo công nghệ MOS cảm ứng ?

5.12. a) Gọi tên 4 mạch con trong hình vẽ mô tả ROM (xem hình B5.12a).

b) Hãy xác định dung lượng của ROM vừa mô tả.

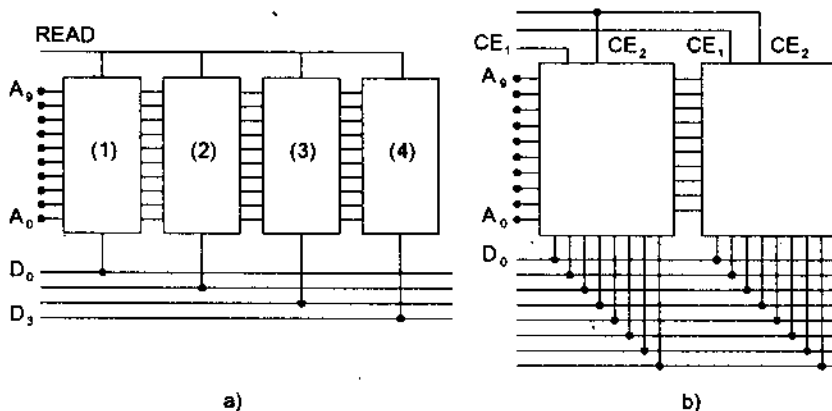
c) Hãy tính dung lượng của bộ nhớ hình B5.12b.



Hình B5.12

5.13. a) Hãy xác định dung lượng của 2 bộ nhớ cho trên hình B5.13a và hình B5.13b dưới đây.

b) Hãy dùng hai đường A_9A_8 để chọn vỏ cho các khối (1) (2) (3) (4), vẽ lại cấu trúc hình B5.13a cho phù hợp.

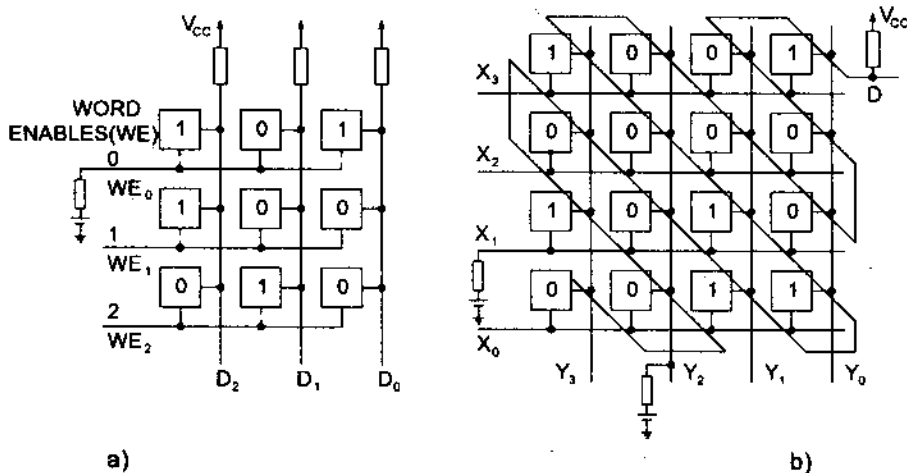


Hình B5.13

5.14. Cho mảng nhớ hình B5.14a và B5.14b

a) Khi đầu vào WE_0 của mạch B5.14a được cấp điện, số nào sẽ xuất hiện tại đầu ra dữ liệu $D_2 D_1 D_0$?

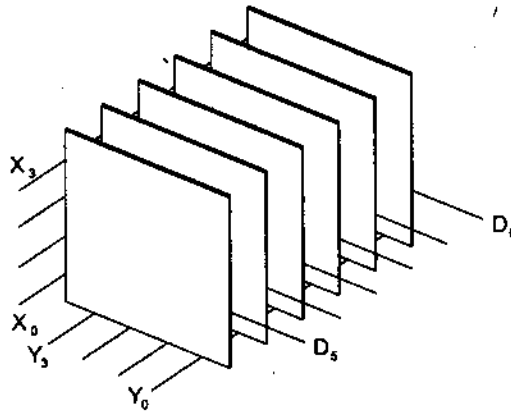
b) Với mạch hình B5.14b khi $X_1 = 1, Y_2 = 1$ tại lối ra sẽ đọc được dữ liệu $D_3 D_2 D_1 D_0$ là gì ?



Hình B5.14

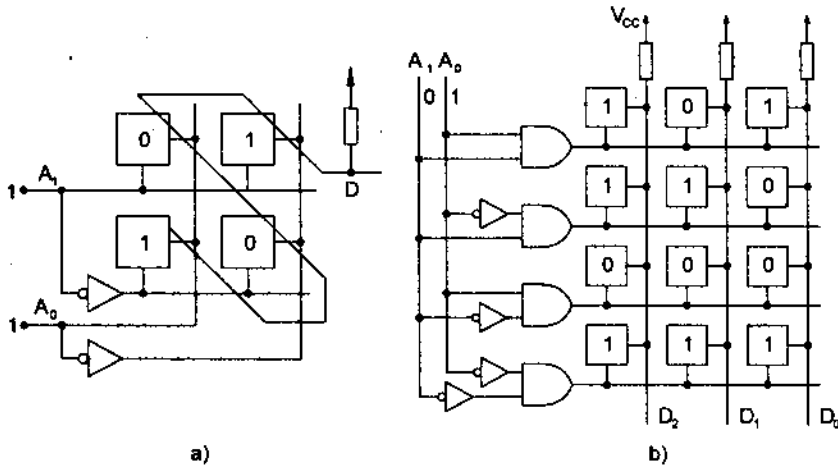
5.15. a) Dung lượng của bộ nhớ cho trên hình B5.15 được tính như thế nào ?

b) Hãy vẽ lại cấu trúc trên mặt phẳng hai chiều.



Hình B5.15

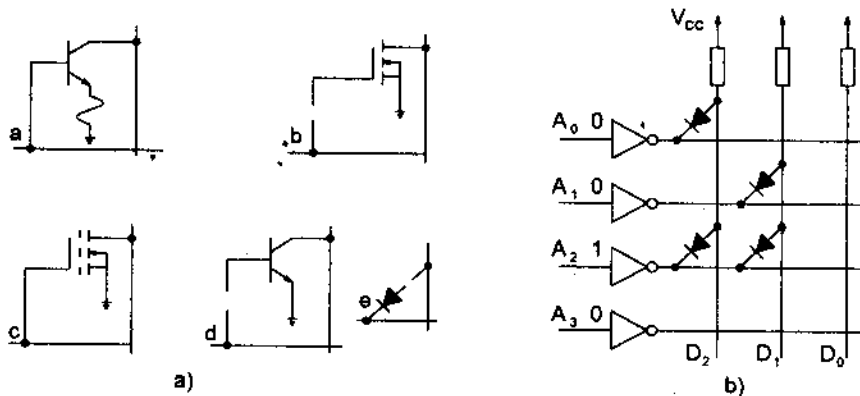
5.16. Trên các bộ nhớ hình B5.16 với các giá trị địa chỉ đã cho trên hình vẽ, xác định giá trị dữ liệu ra tương ứng của chúng.



Hình B5.16

5.17. a) Nhận diện 4 loại ROM cho trên hình B5.17a và nêu tên chúng.

b) Với địa chỉ đã cho $A_0 = A_1 = A_3 = 0, A_2 = 1$, số nào sẽ xuất hiện tại đầu ra D_2, D_1, D_0 trên hình B5.17b.



Hình B5.17

5.18. a) Trazito trường (FET) có cực cửa thả nổi là gì ?

Hãy vẽ sơ đồ ký hiệu quy ước của nó.

Khi nạp dữ liệu cho FET loại này phải làm gì đối với cực cửa thả nổi ?

(b)* Với bộ nhớ hình 5.14b khi cần nạp dữ liệu:

$D_{30} D_{31} D_{32} D_{33} = 1 0 0 1$, $D_{20} D_{21} D_{22} D_{23} = 0 1 1 0$

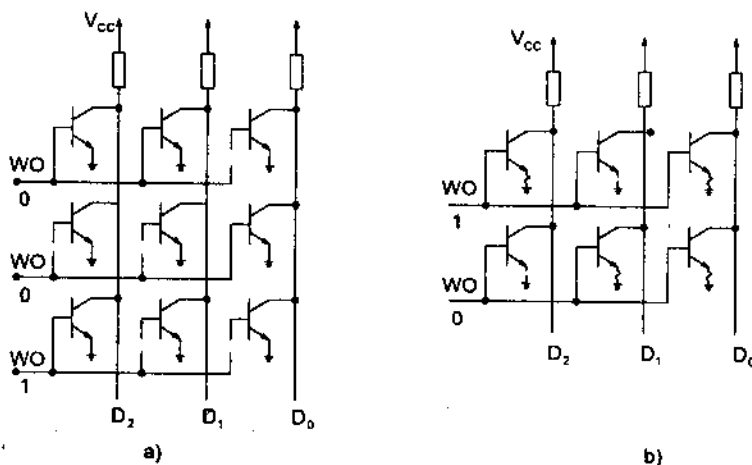
$D_{10} D_{11} D_{12} D_{13} = 1 1 0 1$ và $D_{00} D_{01} D_{02} D_{03} = 0 0 1 1$

Hãy vẽ cấu trúc ma trận các ô nhớ sau khi nạp xong cho EPROM.

c) Với bộ nhớ đã nạp trên, khi cho $WE_2 = 1$, dữ liệu đầu ra là gì ?

5.19. a) Cho bộ nhớ ROM mặt nạ hình B5.19a với tổ hợp tín hiệu cho phép đọc WE_0, WE_1, WE_2 tương ứng bằng 0 0 1, số liệu nào được xuất tại đầu ra của ROM.

b) Câu hỏi tương tự cho PROM hình B5.19b khi đặt $W_0 = 1, W_1 = 0$.



Hình B5.19

5.20. a) Hãy phân biệt điểm khác nhau cơ bản giữa ROM và RAM ? Loại nào có nhiều đầu ra/đầu vào hơn ? Vì sao ?

b) RAM dùng FF chủ tử và dùng tranzitor nhiều emitor có ưu nhược điểm gì? Loại nào thích hợp hơn cho RAM dung lượng lớn ?

c) Hãy phân tích 3 chế độ ghi, chờ và đọc của 1 ô RAM dùng tranzitor nhiều emitor.

5.21. SRAM và DRAM có những đặc điểm gì để phân biệt ?

Ô nhớ dùng tranzitor nhiều cực emitor có dùng trong bộ nhớ DRAM được không? Vì sao ?

RAM có dung lượng lớn thường gặp là SRAM hay DRAM ? Lý do?

5.22. Làm tươi bộ nhớ DRAM là gì ?

Tại sao lại phải làm tươi DRAM ?

Cách tiến hành việc làm tươi như thế nào?

Một DRAM có dung lượng nhớ là 64Kbit có bao nhiêu đầu vào và đầu ra ?
Chứa được bao nhiêu từ nhị phân 8 bit ?

5.23. SRAM 2114 có dung lượng 1024 từ x 4 bit chứa được bao nhiêu byte dữ liệu ? Nếu sử dụng hết các đầu vào địa chỉ của nó thì có thể gán địa chỉ cho bao nhiêu ô nhớ ? Địa chỉ sau có nằm trong số các địa chỉ ô nhớ của SRAM này không?

a) 5FF; b) 3AF ; c) 2AE; d) 7BD; e) A 6 B

5.24. SRAM HM6116 có thời gian truy cập là 120ns nghĩa là thế nào ?

Cần bao nhiêu bit địa chỉ cho nó khi dung lượng là 2048 từ x 8 bit ?

Nó có thể nhớ được bao nhiêu từ 32 byte ?

Nếu ghép song song 8 IC này thì dung lượng bộ nhớ tổng cộng là bao nhiêu ?
Để tránh các địa chỉ trùng lặp trong 8 IC mà ghép sẽ phải làm gì ?

5.25. Dùng phương pháp chọn kênh (Mux) có thể giảm được bao nhiêu đường địa chỉ trong DRAM TMS 4116 ?

DRAM 4164 có dung lượng nhiều gấp đôi của DRAM 4116 có đúng không ?
Việc làm tươi cho nó được thể hiện như thế nào ?

Chu kỳ làm tươi là bao nhiêu ? Thời gian làm tươi và chu kỳ làm tươi có gì khác nhau?

5.26. Hãy phân biệt 3 loại Bus gặp trong việc truyền thông tin giữa các thiết bị số ?

Phương pháp truyền nối tiếp có được sử dụng không ? Vì sao ?

Kết quả tính toán của bộ ALn (đơn vị tính số học logic) sẽ được truyền trên Bus nào?

Phân biệt Bus 1 chiều và Bus 2 chiều, làm thế nào để thực hiện chúng.

5.27. Các vấn đề khó khăn gặp phải khi cho giao tiếp trực tiếp một phần tử số với Bus ?

Biện pháp khắc phục các nhược điểm đó là gì ?

Phân biệt hệ số fanout và fanin của 1 phần tử số ?

Có những cách nào để làm tăng fanout và giảm fanin cho một phần tử số. Tại sao lại phải làm như vậy ?

5.28. Hãy nhận dạng và tìm hiểu hoạt động của mạch hình B5.28.

a) Nêu nhiệm vụ của mạch thông qua việc phân tích chức năng các cổng logic sử dụng và các Bus mà nó tiếp xúc; mô tả lại mạch theo chức năng khối.

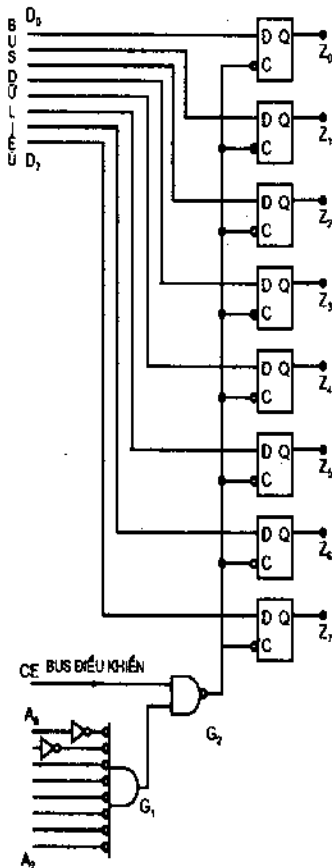
b) Viết các mã địa chỉ được chọn cho mỗi DFF trong mạch đã cho.

5.29. Cho mạch điện hình B5.29

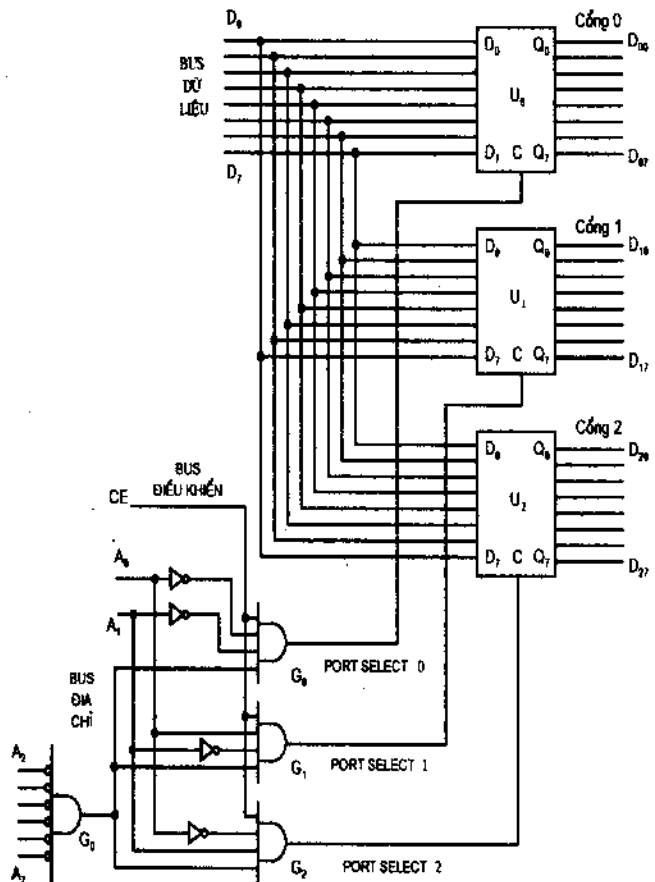
a) Giải thích hoạt động của mạch đã cho.

b) Hãy viết địa chỉ nhóm của IC U_0 , U_1 , U_2 và các địa chỉ cổng ra tương ứng của chúng theo các giá trị của mã $A_0 \rightarrow A_7$.

Trong các tín hiệu vào $D_0 \dots D_7$, $A_0 \dots A_7$ và CE, tín hiệu nào là quan trọng nhất ? Giải thích lý do ?



Hình B5.28



Hình B5.29

MỤC LỤC

	Trang
Lời giới thiệu	3
Mở đầu	4
Chương 1 - Cơ sở kỹ thuật số - 9 tiết (7LT + 2BT)	
1.1. Các hệ thống số đếm	5
1.2. Đại cương về các phép tính số học trong hệ nhị phân	10
1.3. Các phép biến đổi số biểu diễn trong các hệ thống số khác nhau	14
1.4. Các hệ thống mã nhị phân thông dụng	19
1.5. Đại số logic (đại số Boole)	22
1.6. Phương pháp biểu diễn hàm logic và tối thiểu hàm logic	27
Câu hỏi và bài tập chương 1	35
Chương 2 - Các cổng logic cơ bản và mạch điện cổng - 16 tiết (12LT + 4BT)	
2.1. Các tính chất đặc trưng của một cổng logic	40
2.2. Cổng thực hiện hàm cộng logic (cổng OR)	42
2.3. Cổng thực hiện hàm nhân logic (cổng AND)	44
2.4. Cổng thực hiện hàm đảo (phủ định logic - NOT)	45
2.5. Cổng thực hiện hàm logic cộng đảo (cổng NOR)	48
2.6. Cổng logic thực hiện hàm logic nhân - đảo (cổng NAND)	51
2.7. Tính chất đa dụng của cổng NAND và cổng NOR	54
2.8. Một số IC cổng thường gặp	59
2.9. Các ký hiệu logic thay thế	63
2.10. Các mạch điện cổng khác	70
2.11. Ghép nối các cổng IC số	75
Câu hỏi và bài tập chương 2	88
Chương 3 - Các mạch logic tổ hợp (MSI) - 20 tiết (15LT + 5BT)	
3.1. Đặc điểm cơ bản của mạch logic tổ hợp	96
3.2. Các bộ mã hoá	99

3.3. Các bộ giải mã	106
3.4. Bộ chọn dữ liệu (bộ dồn kênh - MUX)	116
3.5. Bộ phân phối dữ liệu (DEMUX)	125
3.6. Bộ so sánh hai số nhị phân cùng cấp (không dấu)	128
3.7. Các bộ cộng số học nhị phân	132
Câu hỏi và bài tập chương 3	144
Chương 4 - Triggers và phần tử logic dãy - 20 tiết (15LT + 5BT)	
4.1. Khái niệm và cấu trúc cơ bản của trigger	148
4.2. RSFF loại đồng bộ	153
4.3. Mạch chốt dữ liệu (LATCH) - trigger trễ hay DFF	155
4.4. Các dạng đặc biệt của RSFF	160
4.5. Trigger chính phụ MSFF (trigger chủ – tớ – Master slave FlipFlop)	162
4.6. Trigger vạn năng JK (JKFF)	173
4.7. Bộ đếm	177
4.8. Một số IC DFF, JKFF, TFF, thanh ghi dịch và bộ đếm thông dụng	189
Câu hỏi và bài tập chương 4	198
Chương 5 - Các bộ nhớ bán dẫn - 10 tiết (9LT + 1BT)	
5.1. Khái niệm chung về bộ nhớ	207
5.2. Bộ nhớ chỉ đọc ROM (READ - ONLY - MEMORY)	210
5.3. Các dạng ROM thường gặp	217
5.4. Bộ nhớ ghi - đọc (RAM)	224
5.5. Khái niệm về BUS	239
Câu hỏi và bài tập chương 5	243
Mục lục	250

Chịu trách nhiệm xuất bản :

Chủ tịch HĐQT kiêm Tổng Giám đốc NGÔ TRẦN ÁI
Phó Tổng Giám đốc kiêm Tổng biên tập NGUYỄN QUÝ THAO

Tổ chức bản thảo và chịu trách nhiệm nội dung :

Chủ tịch HĐQT kiêm Giám đốc Công ty CP Sách ĐH – DN
TRẦN NHẬT TÂN

Biên tập nội dung :

NGÔ THANH BÌNH – TRẦN NGỌC KHÁNH

Trình bày bìa :

TÀO THANH HUYỀN

Sửa bản in :

KIM THƯ

Chế bản :

PHƯƠNG LINH – TRẦN HOÀNG

GIÁO TRÌNH KỸ THUẬT SỐ

Mã số : 7K559y8 – DAI

In 1.500 cuốn (QĐ : 54), khổ 16 x 24cm. In tại Công ty Cổ phần In Phúc Yên.

Địa chỉ : Đường Trần Phú, thị xã Phúc Yên.

Số ĐKKH xuất bản : 04 – 2008/CXB/349 – 1999/GD.

In xong và nộp lưu chiểu tháng 10 năm 2008.



CÔNG TY CỔ PHẦN SÁCH ĐẠI HỌC - DẠY NGHỀ
HEVOBCO
25 HÀN THUYỀN – HÀ NỘI
Website : www.hevobco.com.vn



QUANG TRUNG
HÀ NỘI

**TÌM ĐỌC GIAO TRÌNH DÙNG CHO CÁC TRƯỞNG ĐẠO TẠO HE
TRUNG CẤP CHUYÊN NGHIỆP CỦA NHÀ XUẤT BẢN GIÁO DỤC**

1. An toàn điện	Nguyễn Đình Thắng
2. Kỹ thuật điện	Đặng Văn Đào
3. Máy điện	Nguyễn Hồng Thanh
4. Kỹ thuật lắp đặt điện	Phan Đăng Khải
5. Điện dân dụng và công nghiệp	Vũ Văn Tầm
6. Cung cấp điện	Ngô Hồng Quang
7. Đo lường các đại lượng điện và không điện	Nguyễn Văn Hoà
8. Kỹ thuật điều khiển động cơ điện	Vũ Quang Hồi
9. Điện tử công suất	Trần Trọng Minh
10. Linh kiện điện tử và ứng dụng	Nguyễn Viết Nguyên
11. Điện tử dân dụng	Nguyễn Thanh Trà
12. Kỹ thuật số	Nguyễn Viết Nguyên
13. Kỹ thuật mạch điện tử	Đặng Văn Chuyết
14. Cơ kỹ thuật	Đỗ Sanh
15. An toàn lao động	Nguyễn Thế Đạt
16. Vẽ kỹ thuật	Trần Hữu Quế
17. Vật liệu và công nghệ cơ khí	Hoàng Tùng
18. Dung sai lắp ghép và kỹ thuật đo lường	Ninh Đức Tôn
19. Kỹ thuật sửa chữa ô tô máy nổ	Nguyễn Tất Tiên
20. Công nghệ hàn (lý thuyết và ứng dụng)	Nguyễn Thúc Hà
21. Cơ sở kỹ thuật cắt gọt kim loại	Nguyễn Tiến Lương

*Bạn đọc có thể mua tại các Công ty Sách - Thiết bị trường học ở các địa phương
hoặc các Cửa hàng của Nhà xuất bản Giáo dục :*

Tại Hà Nội : 25 Hàn Thuyên ; 187B Giảng Võ ; 232 Tây Sơn ; 23 Tràng Tiền ;

Tại Đà Nẵng : Số 15 Nguyễn Chí Thanh ; Số 62 Nguyễn Chí Thanh ;

Tại Thành phố Hồ Chí Minh : Cửa hàng 451B - 453, Hai Bà Trưng - Quận 3 ;
240 Trần Bình Trọng - Quận 5.

Tại Thành phố Cần Thơ : Số 5/5, đường 30/4 ;

Website : www.nxbgd.com.vn



Giá: 27.500 đ